

3.2

*IBM Db2 Cloning Tool for z/OS
User's Guide*



Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from the [Db2 Tools Product Documentation page](#).

2021-03-30 edition

This edition applies to Version 3 Release 2 of Db2® Cloning Tool for z/OS® (product number 5655-N15) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2006, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© **Rocket Software Inc. 2006, 2020.**

Contents

| | |
|---|-------------|
| About this information..... | xiii |
| Chapter 1. Db2 Cloning Tool overview..... | 1 |
| What's new in Db2 Cloning Tool..... | 1 |
| New and changed functions..... | 1 |
| Db2 12 function level support..... | 9 |
| Deprecated and removed functions and parameters..... | 10 |
| What does Db2 Cloning Tool do?..... | 12 |
| Cloning definitions..... | 13 |
| Why clone a Db2 subsystem?..... | 13 |
| Can I clone a Db2 subsystem without Db2 Cloning Tool?..... | 13 |
| Db2 support..... | 13 |
| Volume copy products supported by Db2 Cloning Tool..... | 14 |
| Db2 Cloning Tool features and benefits..... | 15 |
| Service updates and support information..... | 15 |
| Product documentation and updates..... | 16 |
| Accessibility features..... | 16 |
| Chapter 2. Preparing to customize Db2 Cloning Tool..... | 19 |
| Set up your environment prior to customization..... | 21 |
| Function authorization requirements..... | 23 |
| Use of the U.S.A. EBCDIC code set..... | 25 |
| Filtering pattern masks..... | 26 |
| Worksheets: Gathering required data set names..... | 27 |
| APF authorizing load libraries..... | 30 |
| Worksheets: Gathering parameter values for Tools Customizer..... | 31 |
| Chapter 3. Starting and preparing Tools Customizer for use..... | 71 |
| Starting Tools Customizer..... | 71 |
| Modifying Tools Customizer user settings..... | 72 |
| Chapter 4. Customizing Db2 Cloning Tool..... | 77 |
| Roadmap: Customizing Db2 Cloning Tool for the first time..... | 77 |
| Roadmap: Customizing a new version of Db2 Cloning Tool from a previous customization..... | 78 |
| Roadmap: Recustomizing Db2 Cloning Tool..... | 79 |
| Specifying the metadata library for the product to customize..... | 80 |
| Discovering Db2 Cloning Tool information automatically..... | 81 |
| Creating and associating Db2 entries..... | 83 |
| Defining parameters..... | 85 |
| Defining Db2 Cloning Tool parameters..... | 85 |
| Defining LPAR parameters..... | 87 |
| Defining Db2 parameters..... | 88 |
| Generating customization jobs..... | 90 |
| Submitting customization jobs..... | 90 |
| Browsing parameters..... | 92 |
| Copying Db2 entries..... | 92 |
| Removing Db2 entries..... | 94 |
| Deleting Db2 entries..... | 94 |
| Displaying customization jobs..... | 95 |
| Maintaining customization jobs..... | 95 |

| | |
|---|------------|
| Using Tools Customizer in a multiple-LPAR environment..... | 96 |
| Chapter 5. After customizing Db2 Cloning Tool using Tools Customizer..... | 97 |
| Reviewing the CKZINI PARMLIB member variables..... | 97 |
| Verifying successful customization..... | 98 |
| Using the Installation Verification Process..... | 98 |
| Adding DB2 Cloning Tool to the DB2 Admin Launchpad..... | 99 |
| Exporting and importing ISPF cloning profiles and subsystem information..... | 99 |
| EXPORT command syntax..... | 100 |
| IMPORT command syntax..... | 101 |
| Migrating table space cloning profiles to application cloning profiles..... | 103 |
| Db2 version migration considerations..... | 105 |
| Chapter 6. Planning for copying and renaming volumes..... | 107 |
| Selection of source and target volumes..... | 107 |
| Volume relationship conflicts..... | 109 |
| Ensure RACF and ACF authorities..... | 109 |
| Data set renaming considerations..... | 109 |
| Target ICF catalog considerations..... | 110 |
| Location of the source and target ICF catalogs..... | 110 |
| Target data set ICF catalog aliases..... | 111 |
| Considerations for DFSMS pervasive encryption..... | 111 |
| Return code choices..... | 111 |
| Cloning a Db2 subsystem..... | 112 |
| Chapter 7. Setting up to copy by volume with FlashCopy, SnapShot, or TimeFinder/Clone..... | 115 |
| Db2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone..... | 115 |
| Steps for volume FlashCopy, SnapShot, or TimeFinder/Clone..... | 116 |
| Chapter 8. Setting up to copy by volume with an onsite mirror tool..... | 119 |
| Db2 Cloning Tool process for user-created target volume clones..... | 119 |
| Steps for user-created target volumes..... | 120 |
| Db2 Cloning Tool process for EMC TimeFinder/Mirror target volume clones..... | 122 |
| Steps for EMC TimeFinder/Mirror target volume clones..... | 122 |
| Chapter 9. Cloning Db2 subsystems..... | 125 |
| Db2 offline cloning..... | 126 |
| Db2 offline cloning procedure..... | 126 |
| Steps for Db2 offline cloning..... | 127 |
| Db2 offline cloning with removal of data sharing members procedure..... | 128 |
| Steps for Db2 offline cloning with removal of data sharing members..... | 129 |
| Db2 offline cloning with target becoming non-data sharing procedure..... | 131 |
| Steps for Db2 offline cloning with target becoming non-data sharing..... | 132 |
| Db2 online cloning..... | 134 |
| Status of transactions in flight..... | 135 |
| Db2 online cloning procedure..... | 135 |
| Steps for Db2 online cloning..... | 137 |
| Db2 online cloning with removal of data sharing members procedure..... | 140 |
| Steps for Db2 online cloning with removal of data sharing members..... | 142 |
| Db2 online cloning with target becoming non-data sharing procedure..... | 146 |
| Steps for Db2 online cloning with target becoming non-data sharing..... | 148 |
| Chapter 10. Subsystem cloning using the Db2 Cloning Tool stored procedure..... | 153 |
| About the Db2 Cloning Tool stored procedure..... | 153 |
| Migrating the stored procedure from Db2 Cloning Tool V3.1..... | 154 |

| | |
|---|-----|
| Steps for cloning a DB2 subsystem using the stored procedure..... | 155 |
| Parameter files and parameter descriptions..... | 155 |
| Calling the stored procedure syntax | 167 |
| Using a REXX EXEC to call the stored procedure..... | 168 |
| Verifying the cloning..... | 169 |
| JCL members generated by the stored procedure..... | 169 |
| Building and running the verified cloning..... | 170 |
| Stored procedure results set..... | 171 |
| Recloning a DB2 system..... | 171 |
| Stored procedure example: Cloning a non-data sharing subsystem..... | 172 |
| Stored procedure example: Cloning by data set..... | 173 |
| Stored procedure example: Cloning from a system-level backup..... | 174 |
| Stored procedure example: Cloning from DB2 system-level backup dump tapes..... | 175 |
| Stored procedure example: Cloning from DB2 system-level backup dump tapes across sysplexes... | 176 |

Chapter 11. Db2 Cloning Tool Table Space Cloning overview..... 179

| | |
|--|-----|
| What does Db2 Cloning Tool Table Space Cloning do?..... | 179 |
| Why you should use Db2 Cloning Tool Table Space Cloning..... | 180 |
| Db2 Cloning Tool Table Space Cloning features and benefits..... | 180 |
| Db2 Cloning Tool Table Space Cloning terminology - Db2 objects vs. VSAM objects..... | 181 |
| Use of the U.S.A. EBCDIC code set..... | 181 |

Chapter 12. Planning for copying Db2 table spaces by data set..... 183

| | |
|--|-----|
| Considerations for in-progress read/write activity or Db2 utilities..... | 183 |
| Considerations for the Db2 Cloning Tool Table Space Cloning cloning process..... | 183 |
| Subsystem copy rules..... | 183 |
| Renaming table spaces and index spaces..... | 183 |
| Implicitly created objects..... | 184 |
| Identity columns..... | 184 |
| LOBs, BLOBs, CLOBs, and DBCLOBs..... | 185 |
| XML considerations..... | 185 |
| TCP/IP considerations..... | 186 |
| APF authorization..... | 186 |
| Ensure RACF and ACF authorities..... | 186 |
| Missing target table spaces and index spaces..... | 186 |
| Considerations for target objects created using DEFINE NO..... | 186 |
| Considerations for DFSMS pervasive encryption..... | 187 |
| Non-SMS managed volumes..... | 189 |
| Space processing order..... | 190 |
| How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces..... | 190 |
| Message output..... | 191 |
| Object mismatch processing in Db2 Cloning Tool Table Space Cloning..... | 191 |
| Object attributes that must be identical between source and target..... | 192 |
| How Db2 Cloning Tool Table Space Cloning checks object compatibility..... | 192 |
| Strategies for resolving object mismatches..... | 194 |
| Tips for specific mismatches..... | 194 |
| Considerations for generating target object DDL using PROCESS-DDL..... | 195 |
| Re-creating existing target objects..... | 196 |
| Sample DDL processing scenarios..... | 198 |
| Considerations for Db2 Version 9.1 and later..... | 200 |
| Considerations for data set renaming..... | 201 |
| Considerations for using FUZZY-COPY..... | 201 |
| Considerations for using catalog prefetch to populate the object cache..... | 201 |
| Considerations for caching indexes..... | 202 |
| Using a local TCP/IP server to populate the target cache for multiple source copy jobs..... | 203 |
| Return code choices..... | 204 |
| Plans and packages..... | 204 |

| | |
|--|------------|
| Dropped and altered tables..... | 204 |
| Copying versioned objects..... | 204 |
| Chapter 13. Db2 Cloning Tool Table Space Cloning function overview..... | 207 |
| Source job overview..... | 207 |
| Target job overview..... | 207 |
| TCP/IP server job overview (optional)..... | 208 |
| Report job overview (optional)..... | 208 |
| Source job details..... | 208 |
| Source job - Init phase I..... | 208 |
| Source job - Discovery phase II..... | 208 |
| Source job - Eligibility phase III..... | 209 |
| Source job - Copy phase IV..... | 210 |
| Target job details..... | 211 |
| TCP/IP server job details (optional)..... | 211 |
| Report job details (optional)..... | 213 |
| CKZFIX job details (optional)..... | 216 |
| Runtime repository functionality overview..... | 216 |
| Chapter 14. Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone..... | 219 |
| Setup for scenario 1: FlashCopy, SnapShot, or TimeFinder/Clone - All target table spaces and index spaces already exist in Db2 catalog..... | 219 |
| Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog..... | 219 |
| Step 2: Set up and execute source job (CKZ1SRC)..... | 220 |
| Step 3: Set up and execute target job (CKZTRG)..... | 223 |
| Setup for scenario 2: FlashCopy, SnapShot, or TimeFinder/Clone - Some or all target table spaces and index spaces do not exist in Db2 catalog..... | 226 |
| Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog (if required).. | 226 |
| Step 2: Set up and execute source job (CKZ2SRC)..... | 227 |
| Step 3 (Optional): Create missing target table spaces and index spaces..... | 231 |
| Step 4: Set up and execute optional FIX job (CKZFIX)..... | 231 |
| Step 5: Set up for a re-run of the source job..... | 232 |
| Step 6: Set up and execute target job (CKZTRG)..... | 233 |
| Chapter 15. Set up procedures for copy by data set for all other methodologies... 237 | 237 |
| Setup for scenario 3: Other copy methodologies - all target table spaces and index spaces already exist in Db2 catalog..... | 237 |
| Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog..... | 237 |
| Step 2: Set up and execute source job (CKZ3SRC)..... | 238 |
| Step 3 (Optional) - Set up and execute stop target job (CKZSTPT)..... | 241 |
| Step 4 (Optional): - Set up and execute stop source job (CKZSTPS)..... | 242 |
| Step 5: Setting up and executing non-Db2 Cloning Tool Table Space Cloning job to execute data set copies outside Db2 Cloning Tool Table Space Cloning..... | 243 |
| Step 6 (Optional): Set up and execute job to re-start source Db2 table spaces and index spaces (CKZSTRS)..... | 243 |
| Step 7: Set up and execute target job (CKZTRG)..... | 244 |
| Setup for scenario 4: Other copy methodologies - some or all target table spaces and index spaces do not exist in Db2 catalog..... | 246 |
| Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog..... | 247 |
| Step 2: Set up source job (CKZ4SRC)..... | 248 |
| Step 3 (Optional): Set up stop target job (CKZSTPT)..... | 252 |
| Step 4 (Optional): - Set up stop source job (CKZSTPS)..... | 252 |
| Step 5: Set up non-Db2 Cloning Tool Table Space Cloning job to execute data set copies outside Db2 Cloning Tool Table Space Cloning..... | 253 |
| Step 6 (Optional): Set up job to re-start source Db2 table spaces and index spaces (CKZSTRS)... | 254 |
| Step 7 (Optional): Create missing target table spaces and index spaces..... | 255 |

| | |
|---|-----|
| Step 8 (Optional): Set up job to make copied VSAM objects available to target DB2 (CKZFIX)..... | 255 |
| Step 9: Set up for a re-run of the source job..... | 256 |
| Step 10: Set up and execute target job (CKZTRG)..... | 256 |

Chapter 16. Procedures for cloning Db2 table spaces and index spaces..... 261

| | |
|---|-----|
| Copy procedure 1: FlashCopy, SnapShot, or TimeFinder/Clone setup - when all target table spaces and index spaces already exist in Db2 catalog..... | 261 |
| Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog..... | 261 |
| Step 2 (Optional): Submit TCP/IP server job..... | 261 |
| Step 3 (Optional): Submit source TCP/IP server job..... | 262 |
| Step 4: Submit source job..... | 262 |
| Step 5: Submit target job..... | 262 |
| Copy procedure 2: FlashCopy, SnapShot, or TimeFinder/Clone setup - when some or all target table spaces and index spaces do not exist in Db2 catalog..... | 262 |
| Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog..... | 263 |
| Step 2 (Optional): Submit TCP/IP server job..... | 263 |
| Step 3 (Optional): Submit source TCP/IP server job..... | 263 |
| Step 4: Submit source job..... | 263 |
| Step 5: Create missing target Db2 table spaces and index spaces - manually or using non-Db2 Cloning Tool Table Space Cloning Tool..... | 264 |
| Step 6: Submit FIX job on target z/OS system (CKZFIX)..... | 264 |
| Step 7: Rerun source job..... | 264 |
| Step 8: Submit target job..... | 264 |
| Copy procedure 3: All other copy methodologies - All target table spaces and index spaces already exist in Db2 catalog..... | 265 |
| Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog..... | 265 |
| Step 2 (Optional): Submit TCP/IP server job..... | 265 |
| Step 3 (Optional): Submit source TCP/IP server job..... | 265 |
| Step 4: Submit source job..... | 266 |
| Step 5: Submit job CKZSTPT to stop target Db2 table spaces and index spaces..... | 266 |
| Step 6: Submit job CKZSTPS to stop source Db2 table spaces and index spaces..... | 266 |
| Step 7: Submit non-Db2 Cloning Tool Table Space Cloning job to copy VSAM objects..... | 266 |
| Step 8: Submit job CKZSTRS to start source Db2 table spaces and index spaces..... | 266 |
| Step 9: Rerun source job..... | 266 |
| Step 10: Submit target job..... | 266 |
| Copy procedure 4: All other copy methodologies - some or all target table spaces and index spaces do not exist in Db2 catalog..... | 267 |
| Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog..... | 267 |
| Step 2 (Optional): Submit TCP/IP server job..... | 268 |
| Step 3 (Optional): Submit source TCP/IP server job..... | 268 |
| Step 3: Submit source job..... | 268 |
| Step 5: Create missing target Db2 table spaces and index spaces - manually or using non-Db2 Cloning Tool Table Space Cloning Tool..... | 269 |
| Step 6: Submit job CKZSTPT to stop target Db2 table spaces and index spaces..... | 269 |
| Step 7: Submit job CKZSTPS to stop source Db2 table spaces and index spaces..... | 269 |
| Step 8: Submit non-Db2 Cloning Tool Table Space Cloning job to copy VSAM objects..... | 269 |
| Step 9: Submit job CKZSTRS to restart source Db2 table spaces and index spaces..... | 269 |
| Step 10: Submit FIX job on target z/OS system (CKZFIX)..... | 269 |
| Step 11: Rerun the source job to prepare target job..... | 270 |
| Step 12: Submit target job..... | 270 |

Chapter 17. Using data masking with table space cloning..... 271

| | |
|---|-----|
| Source job changes..... | 272 |
| Specifying the masking rule..... | 273 |
| MASKRULE(STATIC, <i>numeric</i> " <i>date_or_time</i> " " <i>string</i> ")..... | 273 |
| MASKRULE(MASK, " <i>pattern</i> ", <i>start</i> , <i>end</i>)..... | 274 |
| MASKRULE(PATTERN, " <i>pattern</i> ", " <i>use_sources</i> ")..... | 276 |

| | |
|--|------------|
| MASKRULE(RANDOM, "min", "max", "check_bounds") | |
| MASKRULE(RANDOM, "min_timestamp", "max_timestamp", "min_timezone", "max_timezone", "check_bounds") | 277 |
| MASKRULE(USEREXIT, module) | 279 |
| MASKRULE(SEQUENCE, initial_value, increment) | 279 |
| MASKRULE(SCRAMBLE) | 279 |
| MASKRULE(CURRENT DATE) | |
| MASKRULE(CURRENT TIME) | |
| MASKRULE(CURRENT TIMESTAMP, "timezone") | 280 |
| MASKRULE(CURRENT USER) | 281 |
| Target job changes | 281 |
| Examples of masking jobs | 282 |
| Chapter 18. Using image copies to clone table spaces and index spaces | 289 |
| Steps for cloning from image copies (including index rebuilds) | 290 |
| Selecting a specific image copy to use as the cloning source | 291 |
| Selecting the last image copy to use as the cloning source | 292 |
| Source job changes for cloning from image copies | 292 |
| Chapter 19. Using LOG-APPLY to make consistent copies of table spaces and index spaces | 293 |
| Source job changes | 294 |
| Using LOG-APPLY across multiple LPARs | 294 |
| LOG-APPLY in data sharing environments when one or more members are not available | 295 |
| Target job changes | 296 |
| Chapter 20. Options for rebuilding indexes | 297 |
| Recommendations when rebuilding indexes | 298 |
| Method 1: Intelligent rebuild | 299 |
| Method 2: Using job templates in the source job | 300 |
| Method 3: Using job templates in the target job | 300 |
| Chapter 21. Using job templates | 303 |
| Predefined processing variables | 303 |
| Data set processing variables | 305 |
| Block processing variables | 305 |
| Job templates for utilities | 306 |
| Job templates for cloning consistent FlashCopy data sets to Db2 data sets using IBM Sterling | |
| Connect:Direct | 307 |
| Using the ISPF interface and job templates to clone with IBM Sterling Connect:Direct | 309 |
| Cloning source VSAM files using Sterling Connect:Direct | 310 |
| Chapter 22. Using the ISPF interface | 311 |
| Getting started with the ISPF interface | 311 |
| Configuring job cards | 312 |
| Configuring Db2 subsystems | 312 |
| Working with application cloning profiles | 313 |
| Creating an application cloning profile while setting cloning parameters | 313 |
| Creating an application cloning profile while viewing the profile list | 314 |
| Managing application cloning profiles | 315 |
| Application cloning from Db2 source data sets | 315 |
| Application cloning from image copies | 316 |
| Application cloning using user copies | 316 |
| Application cloning without shared DASD | 317 |
| Selecting the source and target Db2 SSIDs | 318 |
| Selecting source objects | 318 |

| | |
|---|------------|
| Selecting source objects from a list..... | 318 |
| Importing a source object LISTDEF..... | 319 |
| Building application cloning jobs..... | 319 |
| Verifying DD specifications for the cloning jobs..... | 320 |
| Submitting jobs for application cloning from Db2 source data sets..... | 321 |
| Submitting jobs for application cloning from image copies..... | 321 |
| Submitting jobs for application cloning from user copies..... | 321 |
| Submitting jobs for application cloning without shared DASD..... | 322 |
| Chapter 23. Using the classic ISPF interface..... | 323 |
| Getting started with the classic ISPF interface..... | 323 |
| Configuring Db2 subsystems..... | 324 |
| Configuring a subsystem..... | 324 |
| Configuring user settings..... | 326 |
| Setting job card defaults..... | 326 |
| Setting subsystem cloning defaults..... | 326 |
| Setting table space cloning defaults..... | 327 |
| Creating cloning jobs using the interface..... | 334 |
| Creating a profile..... | 335 |
| Subsystem cloning..... | 335 |
| Subsystem cloning steps summary..... | 336 |
| Step summary for subsystem cloning from a system-level backup..... | 336 |
| Select source and target subsystems..... | 337 |
| Add or verify high level qualifiers..... | 338 |
| Specify source and target volume pairings..... | 339 |
| Specify source and target ICF catalogs..... | 339 |
| Specify rename masks for source and target data sets..... | 339 |
| Build the cloning jobs from a profile..... | 340 |
| Submit the jobs..... | 349 |
| Table space cloning..... | 349 |
| Table space cloning steps summary..... | 350 |
| Select the source and target Db2 subsystems..... | 350 |
| Edit the source job..... | 351 |
| Verify the target job settings..... | 352 |
| Verify the report job settings..... | 352 |
| Verify the TCP/IP server job settings..... | 353 |
| Verify the source TCP/IP server job settings..... | 353 |
| Build the table space cloning jobs from a profile..... | 353 |
| Submit the jobs..... | 355 |
| Table space cloning across multiple profiles..... | 355 |
| Chapter 24. Table space cloning using z/OSMF REST services | 357 |
| Sample members and descriptions..... | 357 |
| Variable input file parameters..... | 358 |
| Considerations for multi-line variable values..... | 363 |
| Considerations for multi-line variable values in a JSON request body..... | 364 |
| Configuring and running the workflow..... | 365 |
| Running the workflow via REST services..... | 366 |
| Running the workflow via the z/OSMF web browser-based interface..... | 368 |
| Chapter 25. Cloning table spaces from DB2 Administration Tool..... | 369 |
| Chapter 26. Db2 Cloning Tool Subsystem Cloning commands..... | 371 |
| BACKINFO-REFORMAT..... | 372 |
| BACKINFO-REFORMAT command syntax..... | 373 |
| BACKINFO-REFORMAT command and keyword definitions..... | 374 |
| BACKINFO-REFORMAT step JCL examples..... | 376 |

| | |
|---|-----|
| BCSCLEAN..... | 379 |
| BCSCLEAN command syntax..... | 379 |
| BCSCLEAN command and keyword definitions..... | 380 |
| BCSCLEAN step JCL examples..... | 382 |
| COPY..... | 383 |
| COPY command syntax..... | 386 |
| COPY command and keyword definitions..... | 388 |
| COPY step JCL example..... | 402 |
| COPY-BY-DS..... | 404 |
| Planning for subsystem cloning by data set..... | 404 |
| RENAME considerations for COPY-BY-DS..... | 405 |
| COPY-BY-DS command syntax..... | 407 |
| COPY-BY-DS command and keyword definitions..... | 407 |
| COPY-BY-DS step JCL example..... | 411 |
| COPYCHECK..... | 412 |
| COPYCHECK command syntax..... | 412 |
| COPYCHECK command and keyword definitions..... | 413 |
| COPYCHECK step JCL example..... | 414 |
| DB2ALTERBSDS..... | 414 |
| DB2ALTERBSDS command syntax..... | 415 |
| DB2ALTERBSDS command and keyword definitions..... | 416 |
| DB2ALTERBSDS step JCL example..... | 418 |
| DB2FIX..... | 420 |
| DB2FIX command syntax..... | 421 |
| DB2FIX command and keyword definitions..... | 422 |
| DB2FIX step JCL example..... | 424 |
| DB2GETBACKINFO..... | 424 |
| DB2GETBACKINFO command syntax..... | 425 |
| DB2GETBACKINFO command and keyword definitions..... | 426 |
| DB2GETBACKINFO step JCL examples..... | 428 |
| DB2LGRNXCLEAN..... | 430 |
| DB2LGRNXCLEAN command syntax..... | 430 |
| DB2LGRNXCLEAN command and keyword definitions..... | 430 |
| DB2LGRNXCLEAN step JCL example..... | 431 |
| DB2RBLDBSDS..... | 432 |
| DB2RBLDBSDS command syntax..... | 432 |
| DB2RBLDBSDS command and keyword definitions..... | 433 |
| DB2RBLDBSDS step JCL example..... | 434 |
| DB2SCHEMA-UPDATE..... | 435 |
| DB2SCHEMA-UPDATE command syntax..... | 436 |
| DB2SCHEMA-UPDATE command and keyword definitions..... | 437 |
| DB2SCHEMA-UPDATE step JCL example..... | 439 |
| DB2SETLOG..... | 440 |
| DB2SETLOG command syntax..... | 440 |
| DB2SETLOG command and keyword definitions..... | 440 |
| DB2SETLOG step JCL example..... | 441 |
| DB2SQL..... | 442 |
| DB2SQL command syntax..... | 443 |
| DB2SQL command and keyword definitions..... | 444 |
| DB2SQL step JCL example..... | 447 |
| DB2START..... | 447 |
| DB2START command syntax..... | 449 |
| DB2START command and keyword definitions..... | 449 |
| DB2START step JCL example..... | 451 |
| DB2STOP..... | 452 |
| DB2STOP command syntax..... | 452 |
| DB2STOP command and keyword definitions..... | 453 |
| DB2STOP step JCL example..... | 454 |

| | |
|---|------------|
| DB2UPDATE..... | 454 |
| DB2UPDATE command syntax..... | 455 |
| DB2UPDATE command and keyword definitions..... | 456 |
| DB2UPDATE step JCL example..... | 460 |
| DB2UTILXCLEAN..... | 463 |
| DB2UTILXCLEAN command syntax..... | 464 |
| DB2UTILXCLEAN command and keyword definitions..... | 464 |
| DB2UTILXCLEAN step JCL example..... | 465 |
| Step JCL example to identify objects with UT status..... | 465 |
| DB2XCFCLEAN..... | 466 |
| DB2XCFCLEAN command syntax..... | 466 |
| DB2XCFCLEAN command and keyword definitions..... | 467 |
| DB2XCFCLEAN step JCL example..... | 468 |
| FINDUCATS..... | 468 |
| FINDUCATS command syntax..... | 468 |
| FINDUCATS command and keyword definitions..... | 469 |
| FINDUCATS step JCL example..... | 469 |
| JRNLUPEGRADE..... | 470 |
| JRNLUPEGRADE command syntax..... | 471 |
| JRNLUPEGRADE command and keyword definitions..... | 471 |
| JRNLUPEGRADE step JCL example..... | 472 |
| ONLINECLIP..... | 472 |
| ONLINECLIP command syntax..... | 473 |
| ONLINECLIP command and keyword definitions..... | 473 |
| ONLINECLIP step JCL example..... | 474 |
| RENAME..... | 474 |
| RENAME command syntax..... | 474 |
| RENAME command and keyword definitions..... | 477 |
| RENAME step JCL example..... | 488 |
| RESTORE-FROM-DUMPTAPES..... | 489 |
| RESTORE-FROM-DUMPTAPES command syntax..... | 491 |
| RESTORE-FROM-DUMPTAPES command and keyword definitions..... | 491 |
| RESTORE-FROM-DUMPTAPES step JCL example..... | 495 |
| UCATOPTIONS..... | 496 |
| UCATOPTIONS command syntax..... | 497 |
| UCATOPTIONS command and keyword definitions..... | 497 |
| UCATOPTIONS step JCL example..... | 498 |
| VARYOFF..... | 499 |
| VARYOFF command syntax..... | 499 |
| VARYOFF command and keyword definitions..... | 500 |
| VARYOFF step JCL example..... | 501 |
| VARYON..... | 503 |
| VARYON command syntax..... | 503 |
| VARYON command and keyword definitions..... | 504 |
| VARYON step JCL example..... | 506 |
| VOLOPTIONS..... | 507 |
| VOLOPTIONS command syntax..... | 508 |
| VOLOPTIONS command and keyword definitions..... | 508 |
| VOLOPTIONS step JCL example..... | 510 |
| Chapter 27. Db2 Cloning Tool Table Space Cloning commands..... | 513 |
| COPY..... | 513 |
| COPY command syntax..... | 514 |
| COPY command and keyword definitions..... | 516 |
| OBJECT-TRANSLATE considerations..... | 554 |
| DDL-ATTRIBUTE-CHANGE parameter values..... | 555 |

| | |
|--|-------------|
| GLOBAL command values for EMC TimeFinder/Clone Mainframe Snap Facility data set level support..... | 558 |
| DATASUBTYPE..... | 560 |
| DATASUBTYPE command syntax..... | 560 |
| DATASUBTYPE command & keyword definitions..... | 560 |
| HLQDDDF..... | 561 |
| HLQDDDF command syntax..... | 561 |
| HLQDDDF command and keyword definitions..... | 561 |
| LISTDEF..... | 562 |
| LISTDEF command syntax..... | 564 |
| LISTDEF command and keyword definitions..... | 565 |
| SET..... | 567 |
| SET command syntax..... | 567 |
| SET command and keyword definitions..... | 568 |
| Chapter 28. Troubleshooting..... | 583 |
| Messages..... | 583 |
| Messages and descriptions..... | 583 |
| How to look up message explanations..... | 1080 |
| Gathering diagnostic information..... | 1080 |
| Gathering maintenance level information with the DISPLAY MEPL command..... | 1081 |
| Validating load module contents with MODLEVEL..... | 1082 |
| Tools Customizer troubleshooting..... | 1083 |
| Gathering diagnostic information..... | 1083 |
| Determining the trace data set name..... | 1084 |
| Chapter 29. Tools Customizer reference..... | 1085 |
| Tools Customizer terminology and data sets..... | 1085 |
| Tools Customizer terminology..... | 1085 |
| Data sets that Tools Customizer uses during customization..... | 1087 |
| Chapter 30. Reference..... | 1091 |
| Cloning scenarios..... | 1091 |
| Volume cloning using an interim set of volumes..... | 1091 |
| Volume cloning using multiple interim sets of volumes..... | 1096 |
| Db2 subsystem cloning using data set copy..... | 1103 |
| Db2 subsystem cloning from a Db2 BACKUP SYSTEM backup..... | 1105 |
| Db2 subsystem cloning to a specific point in time from a Db2 BACKUP SYSTEM backup..... | 1115 |
| Db2 subsystem cloning from Db2 BACKUP SYSTEM dump tapes..... | 1128 |
| Db2 subsystem cloning from other system level backups when backup volumes are online..... | 1134 |
| Db2 subsystem cloning from other system level backups when backup volumes are offline..... | 1139 |
| CKZINI customization values..... | 1144 |
| Structure of the CKZINI member..... | 1145 |
| Syntax rules..... | 1145 |
| About Db2 Cloning Tool Table Space Cloning token errors..... | 1146 |
| CKZINI keyword syntax and descriptions..... | 1146 |
| Deallocating target Db2 coupling facility structures..... | 1160 |
| Previous changes to Db2 Cloning Tool..... | 1161 |
| Product legal notices..... | 1183 |
| Index..... | 1187 |

About this information

Db2 Cloning Tool for z/OS (also referred to as Db2 Cloning Tool) makes it easy to quickly clone a Db2 subsystem or a Db2 table space. Db2 Cloning Tool can also be used to clone non-Db2 volumes.

These topics provide instructions for installing, configuring, and using Db2 Cloning Tool.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:

- Plan for the installation of Db2 Cloning Tool
- Install and operate Db2 Cloning Tool
- Customize your Db2 Cloning Tool environment
- Diagnose and recover from Db2 Cloning Tool problems
- Design and write applications for Db2 Cloning Tool
- Use Db2 Cloning Tool with other Db2 products

Tip: To find the most current version of this information, always use [IBM® Knowledge Center](#), which is updated more frequently than PDF books.

Chapter 1. Db2 Cloning Tool overview

Db2 Cloning Tool for z/OS (also referred to as Db2 Cloning Tool) makes it easy to quickly clone a Db2 subsystem or a Db2 table space.

Db2 Cloning Tool can also be used to clone non-Db2 volumes.

What's new in Db2 Cloning Tool

This section describes recent technical changes to Db2 Cloning Tool.

New and changed information is marked like this paragraph, with a vertical bar to the left of a change. Editorial changes that have no technical significance are not marked.

Older changes and enhancements are described in [“Previous changes to Db2 Cloning Tool”](#) on page 1161.

New and changed functions

This topic summarizes the recent enhancements and changes in Db2 Cloning Tool.

2021-03-30

| Description | Related APARs |
|---|---------------|
| <p>Some parameters were removed from table space cloning and are no longer supported. See “Deprecated and removed functions and parameters” on page 10 for a list of the parameters and descriptions of Db2 Cloning Tool behavior when the parameters are not removed from existing jobs. The following topics were updated:</p> <ul style="list-style-type: none">• Removed references to TARGET-JOB-REPAIR-SELECT in the topics “Tips for specific mismatches” on page 194 and “Copying versioned objects” on page 204.• Removed REPLACE-TARGET-DSN from the topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31 and the following source job descriptions and examples:<ul style="list-style-type: none">– “Source job - Copy phase IV” on page 210– “Step 2: Set up and execute source job (CKZ1SRC)” on page 220– “Step 2: Set up and execute source job (CKZ2SRC)” on page 227– “Step 2: Set up and execute source job (CKZ3SRC)” on page 238– “Step 2: Set up source job (CKZ4SRC)” on page 248• Removed REBUILD-COPY-NO-INDEXES from “Method 1: Intelligent rebuild” on page 299 and added optional methods for rebuilding COPY NO indexes.• Removed REBUILD-COPY-NO-INDEXES, ALTER-FOR-XML-LOB-COLUMNS, and REPLACE-TARGET-DSN from the table space cloning topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516.• Removed TARGET-JOB-REPAIR-SELECT and TARGET-JOB-REPAIR-TEST from the table space cloning “SET command syntax” on page 567 and “SET command and keyword definitions” on page 568 topics. | PH24502 |
| When customizing Db2 Cloning Tool, the Db2 APPLCOMPAT bind option supports the value V12R1M507 for package binds in Db2 12 function level 507 subsystems. | PH24502 |

| Description | Related APARs |
|--|---------------|
| <p>Support was added in Db2 Cloning Tool Subsystem Cloning for Remote Pair FlashCopy for XRC. The following topics were updated and messages were added:</p> <ul style="list-style-type: none"> • “Volume copy products supported by Db2 Cloning Tool” on page 14 • “Function authorization requirements” on page 23 • “Parameter files and parameter descriptions” on page 155 • “COPY command syntax” on page 386 • “COPY command and keyword definitions” on page 388 • “COPY-BY-DS command syntax” on page 407 • “COPY-BY-DS command and keyword definitions” on page 407 | PH24502 |
| <p>The RESET-LOGRBA keyword of the table space cloning COPY command has an additional parameter to allow Db2 Cloning Tool to control when the RBA is reset. The topics “COPY command syntax” on page 514, “COPY command and keyword definitions” on page 516, and “CKZINI keyword syntax and descriptions” on page 1146 were updated.</p> | PH24502 |
| <p>The NOT-DB2-SLB keyword was added to better handle subsystem cloning from a system-level backup when backup volumes do not contain SLBs. The topics “DB2ALTERBSDS” on page 414, “DB2ALTERBSDS command syntax” on page 415, and “DB2ALTERBSDS command and keyword definitions” on page 416 were updated and messages were added.</p> | PH24502 |
| <p>The UNLOAD-LOAD-EARLY-VALIDATION parameter was added to the Db2 Cloning Tool Table Space Cloning COPY command to facilitate earlier validation of required templates and control statements in the UNLOAD-LOAD process. The topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516 were updated and messages were added.</p> | PH24502 |
| <p>The TARGET-JOB-MAX-RC parameter was added to the Db2 Cloning Tool Table Space Cloning SET command to override the MAX-RC PARMLIB value when running the target job. The topics “SET command syntax” on page 567 and “SET command and keyword definitions” on page 568 were updated and message “CKZ50017E” on page 898 was added.</p> | PH24502 |
| <p>In the classic ISPF interface, some table space cloning DD names were modified in “Setting default DD specifications for table space cloning” on page 328 and “Table space cloning DD descriptions” on page 330.</p> | PH24502 |
| <p>Information on restricted support in log apply for PBR RPN table spaces was added to the topics “Db2 12 function level support” on page 9, “Set up your environment prior to customization” on page 21, Chapter 18, “Using image copies to clone table spaces and index spaces,” on page 289, Chapter 19, “Using LOG-APPLY to make consistent copies of table spaces and index spaces,” on page 293, and “COPY command and keyword definitions” on page 516.</p> | PH24502 |
| <p>The Db2 Cloning Tool Table Space Cloning option DATA-MOVER PGM(SRCVSCP) has been deprecated. The topic “Deprecated and removed functions and parameters” on page 10 was updated.</p> | None |

2020-06-24

| Description | Related APARs |
|---|---------------|
| Information was added to explain when a combination of END-POINT(TO_CURRENT) and SKIP-LOG-APPLY(Y) can produce consistent table space clones. The topics “Selecting the last image copy to use as the cloning source” on page 292 and “COPY command and keyword definitions” on page 516 were updated. | PH25706 |
| Clarification was added to “Considerations for using catalog prefetch to populate the object cache” on page 201 as to how ENABLE-TARGET-PREFETCH should be set. | None |
| An incorrect CLIST name was corrected in the topics “Verifying successful customization” on page 98 and “Getting started with the ISPF interface” on page 311. | None |
| A SYSUDUMP DD is now provided so that a dump is automatically generated when abends occur in cloning jobs to reduce the time that is required for problem analysis. The topics “Setting default DD specifications for table space cloning” on page 328, “Table space cloning DD descriptions” on page 330, and “Gathering diagnostic information” on page 1080 were updated. | PH24506 |
| When rebuilding indexes for table space cloning, you can use the REBUILD-INDEXES-SORTDEVT keyword to specify the device type for temporary data sets that are to be dynamically allocated by the sort program. The topics Chapter 20, “Options for rebuilding indexes,” on page 297, “Method 1: Intelligent rebuild” on page 299, “SET command syntax” on page 567, and “SET command and keyword definitions” on page 568 were modified. | PH24506 |
| The SPACES-PER-LOG-APPLY-CALL keyword was added to the Db2 Cloning Tool Table Space Cloning COPY command to improve log apply processing under some conditions. The topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516 were updated and message “CKZ503I” on page 681 was added. | PH24506 |
| The topic “Function authorization requirements” on page 23 was updated to add DATAACCESS requirements for cloning objects with XML columns. | None |
| Message “CKZ53551W” on page 907 was updated to clarify recommended settings to avoid conflicts with UNLOAD/LOAD utilities. | PH23774 |
| Message “CKZ61338E” on page 959 was added for inconsistencies in object translate rules with target objects. | PH23774 |
| The Db2 Cloning Tool Subsystem Cloning COPY-BY-DS command RENAME-MASK keyword now supports masking of more than one data set qualifier. The topic “RENAME considerations for COPY-BY-DS” on page 405 was added and the topics “Planning for subsystem cloning by data set” on page 404, “COPY-BY-DS command syntax” on page 407, and “COPY-BY-DS command and keyword definitions” on page 407 were modified. Messages were added and updated in the “Messages” on page 583 topic. | PH23774 |

2020-04-14

| Description | Related APARs |
|---|---------------|
| Db2 Cloning Tool Table Space Cloning supports z/OS Remote Pair FlashCopy for XRC when using DATA-MOVER PGM(ADRDSSU). The topics “Function authorization requirements” on page 23, “COPY command syntax” on page 514, and “COPY command and keyword definitions” on page 516 were updated and messages were added. | PH21116 |

| Description | Related APARs |
|--|---------------|
| All Db2 Cloning Tool packages are now bound using package versions. The topics “Worksheets: Gathering parameter values for Tools Customizer” on page 31 and “Plans and packages” on page 204 were updated. | PH21116 |
| For the installation verification process, when you choose to use Db2 sample tables but your Db2 sample schema name is different from the standard Db2 schema name, you can specify your schema name before building the IVP job. The topics “Worksheets: Gathering parameter values for Tools Customizer” on page 31 and “Using the Installation Verification Process” on page 98 were updated. | PH21116 |
| Support for source and target ICF catalogs that are defined with extended addressability was added to Db2 Cloning Tool Subsystem Cloning. The restriction was removed from the topic “Set up your environment prior to customization” on page 21. | PH21116 |
| The DELETE-MINILOG-DATASETS keyword was added to Db2 Cloning Tool Table Space Cloning to reduce storage usage when cloning across LPARS using target LOG-APPLY. The topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516 were updated. | PH21116 |
| Db2 Cloning Tool Table Space Cloning can automatically start source and target table spaces and index spaces when Db2 utilities such as UNLOAD, LOAD, or REBUILD INDEX are run as part of the cloning process. The topics “How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 190 and “COPY command and keyword definitions” on page 516 were updated and messages were added. | PH21116 |
| When available, the DSNUTILV stored procedure is used when Db2 Cloning Tool Table Space Cloning performs UNLOAD/LOAD functions. The topic “COPY command and keyword definitions” on page 516 was updated and message “CKZ71615I” on page 988 was added. | PH21116 |
| If the Db2 Cloning Tool client and the server operate at different protocol versions, error message “CKZ67235E” on page 965 is displayed to state that the source and target have to be at the same maintenance level. | PH21116 |
| Cloning of archive tables is not supported by Db2 Cloning Tool Table Space Cloning. The topics “Set up your environment prior to customization” on page 21, “Considerations for generating target object DDL using PROCESS-DDL” on page 195, and “LISTDEF” on page 562 were updated. | None |
| Db2 Cloning Tool Table Space Cloning does not support cloning table spaces using the EMC API in an environment where DFSMS pervasive encryption is in use and the EMC SNAP version is later than 8.0.0. The topic “Considerations for DFSMS pervasive encryption” on page 187 was updated and message “CKZ54310I” on page 926 was updated. | PH20535 |
| Message “CKZ64007E” on page 961 was added to provide additional information about OBJECT-TRANSLATE errors. | PH20535 |

2020-02-25

| Description | Related APARs |
|---|----------------------------|
| <p>Information was added to describe how Db2 Cloning Tool Table Space Cloning can automatically process object mismatches and the options you can use to resolve mismatches. The following topics were added or updated: “Object mismatch processing in Db2 Cloning Tool Table Space Cloning” on page 191, “How Db2 Cloning Tool Table Space Cloning checks object compatibility” on page 192, “Object attributes that must be identical between source and target” on page 192, “Strategies for resolving object mismatches” on page 194, and “Tips for specific mismatches” on page 194.</p> | <p>PH18410</p> |
| <p>Existing target objects can now be dropped and re-created as part of DDL generation. You can re-create all target objects or only the target objects with mismatches. The RECREATE-OBJECTS, IGNORE-DROP-OBJECT-DOES-NOT-EXIST, and RECREATE-TABLE-RESTRICTED-ON-DROP keywords were added to the Db2 Cloning Tool Table Space Cloning COPY command. The following topics were added or updated and messages were updated: “Re-creating existing target objects” on page 196, “Sample DDL processing scenarios” on page 198, “How Db2 Cloning Tool Table Space Cloning checks object compatibility” on page 192, “COPY command syntax” on page 514, and “COPY command and keyword definitions” on page 516.</p> | <p>PH18410</p> |
| <p>Db2 Cloning Tool Subsystem Cloning now supports the basic catalog structure (BCS) and the VSAM volume data set (VVDS) being in the extended area of an extended address volume. Restrictions were removed from the topic “Set up your environment prior to customization” on page 21 and messages were deleted, added, or modified.</p> | <p>PH18410</p> |
| <p>The APPLCOMPAT bind option was added to package binds when customizing Db2 Cloning Tool. The topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31 was updated.</p> | <p>PH19508</p> |
| <p>The following changes were made for the new ISPF interface for Db2 Cloning Tool:</p> <ul style="list-style-type: none"> • The minimum region size requirement was updated in “Set up your environment prior to customization” on page 21. • Steps and tasks for the new interface were updated in “Worksheets: Gathering parameter values for Tools Customizer” on page 31 • Added the topic “Migrating table space cloning profiles to application cloning profiles” on page 103 • Added the chapter Chapter 22, “Using the ISPF interface,” on page 311. • Topics that describe how to use the classic ISPF interface that is installed prior to PH09578 were moved to Chapter 23, “Using the classic ISPF interface,” on page 323 and the topics were updated and streamlined. | <p>PH09578 PH18410</p> |
| <p>The requirements for specifying CATWORK-DSN in the Db2 Cloning Tool Subsystem Cloning COPY command were updated. The topics “COPY command syntax” on page 386 and “COPY command and keyword definitions” on page 388 were modified, and messages “CKZ468E” on page 678 and “CKZ469E” on page 678 were added.</p> | <p>PH20535</p> |

| Description | Related APARs |
|---|---------------|
| The SKIP-SYSLGRNX keyword was added to Db2 Cloning Tool Table Space Cloning so that you can turn off SYSLGRNX processing during log apply processing if performance is slow due to a large number of records in a table. The topics Specifying LOG-APPLY defaults , “COPY command syntax” on page 514 , and “COPY command and keyword definitions” on page 516 were updated. | PH18409 |
| The ALLOW-SKIP-INACTIVE-MEMBERS keyword was added to Db2 Cloning Tool Table Space Cloning so that inactive data sharing group members can be excluded from log apply processing. The topic “LOG-APPLY in data sharing environments when one or more members are not available” on page 295 was added, and the topics Specifying LOG-APPLY defaults , “COPY command syntax” on page 514 , and “COPY command and keyword definitions” on page 516 were updated. | PH18409 |
| Instructions were added for using the required template members for the UNLOAD-LOAD DDs in the ISPF interface DD description topic and the COPY command syntax topic. The topics “Table space cloning DD descriptions” on page 330 and “COPY command and keyword definitions” on page 516 were updated. | None |
| Some restrictions for DFSMS pervasive encryption on Db2 Cloning Tool Table Space Cloning cloning from image copies and VSAM-to-VSAM cloning were removed. The topic “Considerations for DFSMS pervasive encryption” on page 187 was updated. | PH18409 |
| The USE-DDL-SQLID parameter was added to the DB2SCHEMA-UPDATE command in Db2 Cloning Tool Subsystem Cloning to specify the SQLID to be used in the CURRENT SQLID parameter of DDL statements. The topics “Parameter files and parameter descriptions” on page 155 , “DB2SCHEMA-UPDATE command syntax” on page 436 , and “DB2SCHEMA-UPDATE command and keyword definitions” on page 437 . | PH14954 |
| The SUPPRESS-GRANTS parameter was added to DDL generation to allow Db2 Cloning Tool Table Space Cloning to clone objects without GRANT statements, thereby removing the need to revoke authorizations on the target system after the cloning. The topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516 were updated. | PH14954 |
| Support for Db2 V12 function level 504: The application cloning process has been enhanced to support Huffman-compressed objects, allow deprecated objects to be cloned between Db2 subsystems, and to create new Db2 Cloning Tool repository objects and convert existing repository objects to UTS table spaces during product customization. The topics “Considerations for generating target object DDL using PROCESS-DDL” on page 195 and “COPY command and keyword definitions” on page 516 were updated, and message “CKZ00314I” on page 703 was updated. | PH14954 |
| Support for Db2 V12 function level 506: Db2 Cloning Tool correctly processes DROP statement enhancements for explicitly created UTS and LOB table spaces. No documentation changes were required. | PH18410 |
| Db2 subsystem information that is entered during product customization now can be exported to a Db2 control data set. When the subsystem information is in the Db2 control data set, you do not need to re-enter that information on the ISPF interface. The topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31 was updated. | PH14954 |

| Description | Related APARs |
|---|---------------|
| For the Db2 Cloning Tool Table Space Cloning VSAM-to-VSAM cloning process, the SKIP-LOG-APPLY(Y) command is required when FUZZY-COPY(Y) to support target object consistency. The topic “COPY command and keyword definitions” on page 516 was updated and messages “CKZ434I” on page 675, “CKZ535128W” on page 913, and “CKZ535129W” on page 913 were updated. | PH17205 |
| Information was added throughout the user guide to better describe the UNLOAD-LOAD process for Db2 Cloning Tool Table Space Cloning. The topics “Table space cloning DD descriptions” on page 330 and UNLOAD-LOAD in “COPY command and keyword definitions” on page 516 were updated. | None |
| Messages related to the ISPF interface, DDL generation, profile migration, and other miscellaneous functions were added. The topic “Messages” on page 583 was updated. | None |

2019-10-11

| Description | Related APARs |
|--|---------------|
| The CKZWMAIN command was added to the list of commands that you might need to add to your ACF2 Command Limiting table. The topic “Set up your environment prior to customization” on page 21 was updated. | None |
| A restriction was added to the UNLOAD-LOAD keyword description that clone instances cannot be processed if their base objects are being processed by UNLOAD-LOAD. The UNLOAD-LOAD description in the “COPY command and keyword definitions” on page 516 topic was updated. | None |
| The process for cloning universal table spaces (UTS) was improved. No documentation changes were required other than new messages that were added to the topic “Messages” on page 583. | PH15193 |
| The table space cloning process has been enhanced to allow applying logs to indexes. The ISPF interface topics COPY command defaults and Specifying LOG-APPLY defaults were updated, as well as the topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516. | PH14349 |
| The subsystem cloning COPY-BY-DS command was enhanced so that a FlashCopy target volume can be a Peer-to Peer Remote Copy (PPRC) primary volume. The topics “COPY-BY-DS command syntax” on page 407, “COPY-BY-DS command and keyword definitions” on page 407, and the stored procedure topic “Parameter files and parameter descriptions” on page 155 were updated. | PH13041 |
| Db2 Sort can be specified as an option for the LOG-APPLY SORT-PROGRAM keyword. The ISPF interface topic SET command defaults and the topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516 were modified. | PI97593 |
| A task was added to Tools Customizer to free plans and packages before a rebind. The topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31 was updated. | PH12071 |
| Storage class and management class parameters were added to the subsystem cloning COPY-BY-DS command in the Db2 Cloning Tool Subsystem Cloning stored procedure. The topic “Parameter files and parameter descriptions” on page 155 was updated. | PH12071 |

| Description | Related APARs |
|---|-------------------------|
| <p>PTF PH09578 contains extensive changes to the Db2 Cloning Tool ISPF panels, specifically for table space cloning (now called application cloning). Existing Db2 Cloning Tool customers can bypass installation of the new ISPF interface that is implemented in PTF PH09578 by following the instructions as noted in the step and task descriptions contained in Table 16 on page 56, in the topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31. New customers can run the existing ISPF interface by modifying the CLIST after the jobs have been run; see the instructions in Table 16 on page 56.</p> | <p>PH09578, PH13598</p> |
| <p>The SLB-START-NOT-CREATED parameter was added to Db2 Cloning Tool Subsystem Cloning to allow you to specify the return code when an SLB start record is not created. The topics “DB2ALTERBSDS command syntax” on page 415 and “DB2ALTERBSDS command and keyword definitions” on page 416 were updated, and message CKZ27124I was updated.</p> | <p>PH12071</p> |
| <p>The subsystem cloning COPY-BY-DS command can now be configured so that the STORCLAS and MGMTCLAS can be the same as the source data sets, be specified by the user, or be null. The topics “COPY-BY-DS command syntax” on page 407 and “COPY-BY-DS command and keyword definitions” on page 407 were updated messages were added to “Db2 Cloning Tool messages” on page 654 to support this feature.</p> | <p>PH10393</p> |
| <p>The topic “Job templates for cloning consistent FlashCopy data sets to Db2 data sets using IBM Sterling Connect:Direct” on page 307 was updated to add instructions to verify a step in the template job.</p> | <p>PH10393</p> |
| <p>Support was added for subsystem cloning from and to subsystems with catalogs that are enabled for record-level sharing (RLS). No user interface changes were made, but messages were added to “Db2 Cloning Tool messages” on page 654 to support this feature.</p> | <p>PH09578</p> |
| <p>Db2 Cloning Tool Table Space Cloning now uses the Db2 sliding scale algorithm when allocating data sets that are used in log apply functions. No user interface changes were made, but messages CKZG4600E through CKZG4630E were added to “Db2 Cloning Tool messages” on page 654 to support this feature.</p> | <p>PH09283</p> |
| <p>The WAIT-TAPE-ALLOC keyword was added to the RESTORE-FROM-DUMPTAPES command to allow you to specify how long Db2 Cloning Tool Subsystem Cloning should wait for an available tape drive if all tape drives are busy at the start of the job. The topics “RESTORE-FROM-DUMPTAPES command syntax” on page 491, “RESTORE-FROM-DUMPTAPES command and keyword definitions” on page 491, “Parameter files and parameter descriptions” on page 155, and “Stored procedure example: Cloning from Db2 system-level backup dump tapes across sysplexes” on page 176 were updated and messages were added to “Db2 Cloning Tool messages” on page 654 to support this feature.</p> | <p>PH08154</p> |
| <p>Clarifications were added for the DEFAULT-SQLID parameter for Db2 Cloning Tool Table Space Cloning. The ISPF interface topic SET command defaults and the topics “LISTDEF command and keyword definitions” on page 565 and “SET command and keyword definitions” on page 568 were updated, and messages “CKZB437E” on page 1031 and “CKZB438E” on page 1031 were added.</p> | <p>None</p> |

| Description | Related APARs |
|---|---------------|
| <p>The SYSPLEX_GROUPNAME_ALL parameter was added to Db2 Cloning Tool Subsystem Cloning to allow you to define the scope of the z/OS MVS™ ROUTE command. The following topics were updated and messages were added for this feature: “Worksheets: Gathering parameter values for Tools Customizer” on page 31, “Parameter files and parameter descriptions” on page 155, “RENAME command syntax” on page 474, “RENAME command and keyword definitions” on page 477, “RESTORE-FROM-DUMPTAPES command syntax” on page 491, “RESTORE-FROM-DUMPTAPES command and keyword definitions” on page 491, “VARYOFF” on page 499, “VARYOFF command syntax” on page 499, “VARYOFF command and keyword definitions” on page 500, “VARYON” on page 503, “VARYON command syntax” on page 503, “VARYON command and keyword definitions” on page 504, “CKZINI keyword syntax and descriptions” on page 1146.</p> | PH07643 |

Db2 12 function level support

When you activate new Db2 12 function levels in a Db2 subsystem or data sharing group, enhancements might become available that impact Db2 Cloning Tool.

The levels of function level support are defined as follows:

Tolerated

The product works as it did on a previous release or function level of Db2 for z/OS, but it does not support the new features of this function level.

Supported

The product supports most, but not necessarily all, of the new function-level features that IBM deems the most significant for this product.

The following function levels are tolerated or supported by Db2 Cloning Tool and are provided with the corresponding PTF, if any. For more information about a specific function level, click the link in the "Db2 function level" column.

| Db2 12 function level | Toleration PTF | Support PTF |
|-----------------------|--------------------------------------|---|
| FL507 | Either UI66072 + UI66073, or UI67229 | UI70843 When customizing Db2 Cloning Tool, the Db2 APPLCOMPAT bind option supports the value V12R1M507 for package binds. |
| FL506 | Either UI66072 + UI66073, or UI67229 | Either UI66072 + UI66073, or UI67229 Existing target objects can be dropped and re-created as part of DDL generation. See the enhancement description (PH18410) in “New and changed functions” on page 1 . |
| FL505 | UI66072 + UI66073 | UI66072 + UI66073 |

Table 1. Db2 Cloning Tool PTFs in support of Db2 12 function levels (continued)

| Db2 12 function level | Toleration PTF | Support PTF |
|-----------------------|----------------|---|
| FL504 | UI60399 | UI66072 + UI66073 See Db2 12 function level 504 enhancements in “ New and changed functions ” on page 1. |
| FL503 | UI60399 | UI60399 |
| FL502 | UI43618 | UI60399 See Db2 12 function level 502 enhancements in “ Previous changes to Db2 Cloning Tool ” on page 1161. |
| FL501 | UI43618 | UI43618 |
| FL500 | UI36973 | UI43618 See Db2 12 support enhancements in “ Previous changes to Db2 Cloning Tool ” on page 1161. Restriction: Db2 table space cloning does not support log apply when cloning partition-by-range (PBR) relative page number (RPN) table spaces with these parameters in the COPY command: <ul style="list-style-type: none"> • DATA-MOVER PGM(SRCIMCPY) • DATA-MOVER PGM(SRCVSCP) • LA-ENABLE(Y) |

Deprecated and removed functions and parameters

This topic summarizes the Db2 Cloning Tool functions and parameters that were recently deprecated or removed.

Deprecated functions

Certain capabilities that Db2 Cloning Tool supports are deprecated, meaning that their use is discouraged. Although they remain supported except as noted below, support is likely to be removed in the future.

| Deprecated function | Recommended alternative | Target removal date |
|---|---|---------------------|
| Db2 Cloning Tool Table Space Cloning option DATA-MOVER(SRCVSCP) | For most cases, specify DATA-MOVER PGM (ABRDSSU) and FASTREP(PREF). This setting specifies that FlashCopy is to make the copies when FlashCopy is available. The determination is made at run time. If you cannot use FlashCopy, set FASTREP to NONE to allow ABRDSSU to choose a host-based copy option. | -- |

Removed functions and parameters

The following functions and parameters are no longer supported:

| Description | Related APARs |
|---|---------------|
| <p>The following Db2 Cloning Tool Table Space Cloning parameters are obsolete and have been removed to avoid potential issues with target objects after cloning. If your existing source job contains these parameters, you should remove them from the job.</p> <ul style="list-style-type: none"> • ALTER-FOR-XML-LOB-COLUMNS • REBUILD-COPY-NO-INDEXES • REPLACE-TARGET-DSN • TARGET-JOB-REPAIR-SELECT • TARGET-JOB-REPAIR-TEST <p>If these unsupported parameters are not removed from the source job, Db2 Cloning Tool Table Space Cloning behavior is as follows:</p> <ul style="list-style-type: none"> • If you do not remove ALTER-FOR-XML-LOB-COLUMNS from the source job, Db2 Cloning Tool ignores the value of ALTER-FOR-XML-LOB-COLUMNS. During the creation of target objects by DDL generation, ALTER statements are used to create XML DOCID and LOB ROWID columns in the correct order. • If you do not remove REBUILD-COPY-NO-INDEXES from the source job, Db2 Cloning Tool ignores the value of REBUILD-COPY-NO-INDEXES. If you are cloning from image copies, rebuild of indexes that are defined with the COPY NO attribute are automatically done either in the target job (when REBUILD-INDEXES-EXECUTE(Y) is specified in the SET command of the source job), or in a generated REBUILD INDEX job (when either CKZINTRB DD is specified in the target job JCL or the SET command parameter TARGET-JOB-INDEX-REBUILD-DDN is specified in the source job). • If you do not remove REPLACE-TARGET-DSN from the source job, Db2 Cloning Tool ignores the value of REPLACE-TARGET-DSN and automatically replaces existing target VSAM data sets when cloning in non-simulate mode and DATA-MOVER PGM is not NONE. • If you do not remove TARGET-JOB-REPAIR-SELECT from the source job, Db2 Cloning Tool ignores the value of TARGET-JOB-REPAIR-SELECT. If TARGET-JOB-REPAIR-EXECUTE(Y) is specified or the TARGET-JOB-REPAIR-DDN is provided, a REPAIR utility automatically resolves mismatches. • If you do not remove TARGET-JOB-REPAIR-TEST from the source job, Db2 Cloning Tool ignores the value of TARGET-JOB-REPAIR-TEST. | PH24502 |
| <p>For Db2 Cloning Tool Subsystem Cloning, the NUM-DS keyword of the COPY-BY-DS command is obsolete. If your existing JCL contains the NUM-DS keyword, it is ignored. The COPY-BY-DS command no longer restricts the number of total data sets per RENAME-MASKS mask.</p> | PH23774 |
| <p>DDF connections via the REMOTE-CONNECT-TYPE keyword in Db2 Cloning Tool Table Space Cloning are no longer supported. DDF connections are replaced with TCP/IP connections to provide better performance for cross-LPAR cloning. If your existing JCL uses the REMOTE-CONNECT-TYPE(D) command, change the command to REMOTE-CONNECT-TYPE(T). If Db2 Cloning Tool Table Space Cloning encounters REMOTE-CONNECT-TYPE (D) in existing JCL, REMOTE-CONNECT-TYPE (T) is used. Multiple topics and messages were updated throughout the user guide to remove references to the DDF remote connection type, and messages were added.</p> | PH21116 |

| Description | Related APARs |
|--|---------------|
| For Db2 Cloning Tool Table Space Cloning, the EXTEND-TARGET-PBG-TABLESPACE command has been removed. When a target partition-by-growth (PGB) table space has less partitions than the source PBG table space, ALTER ADD PARTITION syntax is automatically generated. If your existing JCL contains these parameters, you should remove them from the JCL. | PI92866 |
| <p>The following Db2 Cloning Tool Table Space Cloning parameters are obsolete. If your existing JCL contains these parameters, you should remove them from the JCL.</p> <ul style="list-style-type: none"> • COPY command V7-MIGRATED-OBJECTS-PRESENT parameter • COPY command parameter SOURCE-PREFETCH-DATABASE-LIST in the CATALOG-PREFETCH parameter | PI92862 |

What does Db2 Cloning Tool do?

Db2 Cloning Tool has two components: Db2 subsystem cloning and Db2 table space cloning.

Db2 subsystem cloning:

- Clones automatically by using IBM FlashCopy, STK SnapShot, or EMC TimeFinder/Clone Mainframe Snap Facility's volume level support
- Uses any volume fast replication or onsite mirror tool

The cloned Db2 subsystem can be accessed from the same z/OS system. This increases productivity in several ways:

- Significantly reduces production online downtime and the costs associated with cloning a Db2 subsystem with traditional tools.
- Provides more availability and services to customers because Db2 no longer needs to be shut down or conditioned the long traditional way.
- Uses less personnel time to clone a Db2 subsystem – what used to take hours or days now takes just minutes.
- Allows management of larger storage environments with the same staff.
- Provides quicker throughput and faster turnaround time.
- Provides virtually 24x7 access to data.
- Creates fast quality assurance and/or test environments.

Volume fast replication tools allow a group of data to be replicated within minutes to provide duplicate environments. However, these clones have an inherent problem: the internal volume name, volume internals, and all data set names reflect the source volume name.

Db2 Cloning Tool quickly solves this problem. Once the data is cloned or replicated, Db2 Cloning Tool renames and catalogs the data sets on the cloned volumes, fixes the volume internals, and updates the Db2 internals so the cloned Db2 subsystem can be accessed from the same MVS system. Db2 Cloning Tool provides a vast improvement over existing methods. It supports entire Db2, Db2 PeopleSoft, and Db2 SAP subsystems in either online or offline mode.

Db2 table space cloning:

- Clones automatically by using IBM FlashCopy, STK SnapShot, or EMC TimeFinder/Clone Mainframe Snap Facility's data set level support
- Uses any data set copy, fast or slow

The cloned Db2 table spaces can be cloned within the same or different z/OS system. This increases productivity in several ways:

- Uses less personnel time to copy Db2 table spaces and associated manual tasks – what used to take hours now takes just minutes.
- Provides quicker throughput and faster turnaround time.
- Provides virtually 24x7 access to data.
- Creates fast refreshes of quality assurance and/or test environments

The remainder of this introductory topic addresses only Db2 subsystem cloning using volume copies. For more information about Db2 Cloning Tool table space cloning, refer to [Chapter 11, “Db2 Cloning Tool Table Space Cloning overview,”](#) on page 179.

Cloning definitions

A clone is an exact copy, indistinguishable from the original.

Cloning is the act of replicating data, making it accessible, and then using the replica in lieu of the original data for other purposes. Replication tools clone the data by volume, and Db2 Cloning Tool makes the clone accessible.

Why clone a Db2 subsystem?

There are several reasons to clone a Db2 subsystem.

Clone a Db2 subsystem:

- To create a production quality assurance environment
- To move a group of end users to the cloned Db2 subsystem to lessen the performance impact on the production system
- To give end users access to an application that is updated on a continual basis
- To allow developers to begin changing the application for the next phase of code updates
- To test new functions and features of SAP, or PeopleSoft
- To run an online inquiry while batch runs
- For data mining
- For data warehousing

Can I clone a Db2 subsystem without Db2 Cloning Tool?

A Db2 subsystem can be cloned without using Db2 Cloning Tool but it is a complicated process that can take days, and requires using target volumes on a separate LPAR.

By using Db2 Cloning Tool, you can clone a Db2 subsystem within minutes instead of hours or days. Also, Db2 Cloning Tool does not require a separate LPAR.

Db2 support

A Db2 subsystem can be cloned in either an offline environment (Db2 is stopped and started) or an online environment (using Db2 SUSPEND and RESUME, or via consistent FlashCopy, SnapShot, or TimeFinder/Clone, or via consistent split or break mirror). There is no requirement for a separate LPAR.

Db2 Cloning Tool updates Db2 internal control information in the BSDS, directory, and Db2 catalog to reflect the target Db2 subsystem names.

Db2 data sharing groups and non-data sharing Db2 subsystems are supported. When cloning a Db2 data sharing group, you can optionally reduce the number of Db2 data sharing members, or go from data sharing to non-data sharing.

Volume copy products supported by Db2 Cloning Tool

Db2 Cloning Tool will rename and catalog data sets on target volumes created with any type of replication mechanism, where target volumes are exact replicas of source volumes.

If target volumes still have the source volume serial number (VOLSER), and are varied offline, Db2 Cloning Tool can re-label and vary the target volumes online.

Restriction: "Exact Replicas": The only exception to a target volume being an exact replica of its source volume counterpart is the internal volume serial number. Copies already partially modified are not acceptable – such as volume snaps via SIBBATCH where the SYS1.VVDS and SYS1.VTOCIX names may have already been modified.

Storage blades

The fast replication copy services outlined in this section can be used to create the volume copies used for Db2 system clones, or to create the data set copies used for table space and index space refreshes. A storage blade represents fast replication copy services invoked directly by Db2 Cloning Tool.

Db2 Cloning Tool executes the DFSMSdss blade to issue IBM FlashCopy or STK SnapShot copy commands either by volume or by data set. When cloning using ADRDSSU, Db2 Cloning Tool specifies the DUMPCONDITIONING keyword. When cloning Db2 systems or table spaces that reside on EMC DASD, Db2 Cloning Tool uses the EMC API to invoke TimeFinder/Clone to copy the data by volume or by data set, and specifies the COPYVOLID=N and VARY_OFFLINE=N keywords. When cloning Db2 subsystems that reside on EMC DASD and use TimeFinder/Mirror, or that reside on Hitachi Storage Systems, an appropriate process is performed before Db2 Cloning Tool cloning automation is invoked and a list of copied storage volumes are passed to Db2 Cloning Tool for use in later processing steps.

IBM storage blades

- IBM DFSMSdss blade
- ADRDSSU utility invoked
- Fast replication (preferred)
- By volume or data set
- FlashCopy V2 (IBM, EMC, HDS)
- SnapShot (STK, Ramac)

EMC storage blades

- EMC TimeFinder
- TimeFinder/Clone Mainframe Snap Facility's volume level support
- EMC Consistency Technology
- TimeFinder/Clone Mainframe Snap Facility's data set level support

IBM FlashCopy, STK SnapShot, and EMC TimeFinder/Clone allow the creation of what appears to be a copy of a volume, in a short period of time. The target volume of a copy will be a mirror image of the source volume, with the exception of the internal 'VOLSER', which remains as is, if the COPYVOLID option is not specified. Db2 Cloning Tool requires the target VOLSER to remain as is, which allows the target volume to remain varied online after the copy initiation, and therefore negates the need for a CLIP (re-label) and VARY ON.

Because the logical image of a target volume may be modified before the copy is actually complete, the Db2 Cloning Tool rename step may be executed while the background copy is still in progress. Likewise, should the Db2 Cloning Tool process complete before a volume is completely copied, the application may also begin using volumes before the background copy is complete.

Onsite mirror tools by volume

Other cloning mechanisms, such as IBM PPRC/XRC, Hitachi ShadowImage, EMC TimeFinder/Mirror, Softek Replicator, or Innovation Data Processing FDRPAS, allow the creation of a point-in-time image of a

volume by establishing a mirror, and then splitting the mirror once the target volume is synchronized with the source volume. At the time of this publication, establishing and splitting mirrors must be accomplished with user-provided steps prior to execution of Db2 Cloning Tool.

Db2 Cloning Tool features and benefits

Db2 Cloning Tool provides solutions to many different types of problems.

Db2 Cloning Tool offers several unique and significant features that you can use to quickly clone your Db2 data for use in testing or other requirements:

- Quickly fixes volume conflicts (VTOC, VTOCIX and VVDS) and then renames and re-catalogs the target data sets to solve the data access problem
- Offers extended rename capability to support Db2 log and BSDS desired names
- Db2 online or offline cloning in minutes instead of days, without a separate image
- Automatic updating of Db2 internals to reflect renamed data sets
- Db2 data sharing
- Db2 data sharing many to less members
- Db2 support for either Db2 data sharing or from data sharing to non-data sharing for maximum flexibility
- Provides automatic pairing of volume characteristics (SMS and non-SMS, or by device size) that DFSMSdss doesn't do
- Allows FlashCopy, SnapShot, or TimeFinder/Clone by VOLSER masks or entire storage groups or any combination to eliminate the requirement for individual volume specification
- Provides extensive SMS options that enable you to determine how the SMS class constructs will be applied to your cloned data sets to ensure they are managed correctly
- Issuance of DFSMSdss commands or EMC TimeFinder/Clone commands within the product for ease of use
- Source volume ICF catalog information collection concurrent with FlashCopy, SnapShot, or TimeFinder/Clone initiation for rename integrity
- Early resumption of source volume activity to reduce outages
- Enhanced data set rename masking characters for flexibility
- User options to decide disposition of 'abnormal' data sets and catalog entries
- Multivolume data set and VSAM sphere integrity checks to prevent orphaned data
- Rename into existing populated ICF catalog(s) or a specific one
- Faster cataloging than conventional means
- Test for termination of FlashCopy, SnapShot, or TimeFinder/Clone relationships
- Mechanism to remove orphaned catalog entries from previous executions
- Simulate modes for most commands

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Tools_for_z~OS

Product documentation and updates

Db2 Tools information is available at multiple places on the web. You can receive updates to Db2 Tools information automatically by registering with the IBM My Notifications service.

Information on the web

The most current version of this information is available on IBM Knowledge Center:

<http://www.ibm.com/support/knowledgecenter>

A PDF version of this information is available on the Db2 Tools Product Documentation web page; however, IBM Knowledge Center is updated more frequently than PDF books. The Db2 Tools Product Documentation web page is located at:

<http://www.ibm.com/support/docview.wss?uid=swg27020910>

IBM Redbooks® publications that cover Db2 Tools are available from the following web page:

<http://www.redbooks.ibm.com>

The IBM Information Management System website shows how IT organizations can maximize their investment in Db2 databases while staying ahead of today's top data management challenges:

<https://www.ibm.com/analytics/us/en/db2/db2-for-zos/>

Receiving documentation updates automatically

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The Db2 Tools option is located under **Software > Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

How to send your comments

Your feedback helps IBM to provide quality information. Send any comments that you have about this book or other Db2 Tools documentation to comments@us.ibm.com. Include the name and version number of the product and the title and number of the book. If you are commenting on specific text, provide the location of the text (for example, a chapter, topic, or section title).

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in this product enable users to perform the following activities:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.

- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*
 - *z/OS TSO/E Primer*
 - *z/OS TSO/E User's Guide*

These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Chapter 2. Preparing to customize Db2 Cloning Tool

Before you start to customize Db2 Cloning Tool for the first time, determine all of the customization values that you need to specify during the customization process, and familiarize yourself with all of the customization tasks.

Checklist for customization tasks

The following checklist lists and describes each significant customization step. Use this checklist to guide you through the entire customization process.

Tip: Print the following checklist and the data set names and parameter values worksheets. Use the worksheets to record your values, and refer to them during the customization process.

| Task | Link to detailed instructions | Status |
|--|--|--------|
| Tools Customizer basics | | |
| Before you begin the customization process, familiarize yourself with Tools Customizer terminology, data sets, and other basic information about Tools Customizer. | “Tools Customizer terminology” on page 1085 and “Data sets that Tools Customizer uses during customization” on page 1087 | |
| Software requirements | | |
| Verify that your environment meets the minimum software requirements. To install and use Db2 Cloning Tool, your environment must be running a supported version of the z/OS operating system and of Db2 for z/OS. Additionally, certain levels of maintenance must be applied. | “Verify that your environment meets software requirements” on page 21 | |
| SMP/E installation | | |
| Verify that Db2 Cloning Tool was installed correctly. Db2 Cloning Tool is installed by using standard SMP/E processing. | “Verify that Db2 Cloning Tool has been installed successfully” on page 22 | |
| Verify that Tools Customizer for z/OS was installed correctly. Tools Customizer for z/OS is installed by using standard SMP/E processing. | “Verify that Tools Customizer for z/OS has been installed successfully” on page 22 | |
| Security requirements | | |
| Confirm that you have the required authorizations to use Db2 Cloning Tool. | “Verify that your environment meets security requirements” on page 22 | |
| Function authorization requirements | | |
| Review the level of authority required to perform certain Db2 Cloning Tool functions. | “Function authorization requirements” on page 23 | |
| Use of the U.S.A. EBCDIC code set | | |

| Task | Link to detailed instructions | Status |
|---|--|--------|
| Db2 Cloning Tool uses the U.S.A. EBCDIC code set for specification and display of EBCDIC characters and for the extended ACS masking characters used for filtering. If the code tables used by your installation are different, then you may need to specify characters particular to your code table. | “Use of the U.S.A. EBCDIC code set” on page 25 | |
| Filtering pattern masks | | |
| Many Db2 Cloning Tool commands allow specification of powerful filtering masks. These filtering pattern masks can be specified with extended ACS masking characters, as specified in the table. Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set. | “Filtering pattern masks” on page 26 | |
| Gather data set names | | |
| During the customization process, you must specify data set names for Tools Customizer, Db2 Cloning Tool, and several other items. | “Worksheets: Gathering required data set names” on page 27 | |
| APF authorization | | |
| The SCKZLOAD library must be APF authorized. | “APF authorizing load libraries” on page 30 | |
| Gather parameter values | | |
| During the customization process, you must specify parameter values for Db2 Cloning Tool, for Db2, and for your LPAR. | “Worksheets: Gathering parameter values for Tools Customizer” on page 31 | |
| Customize Db2 Cloning Tool | | |
| Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel. | “Starting Tools Customizer” on page 71 | |
| Set up Tools Customizer user settings. If you are running Tools Customizer for the first time, you must modify several user settings for your environment. Otherwise, if the user settings that you have already established are still appropriate, skip this step. | “Modifying Tools Customizer user settings” on page 72 | |
| Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing. | | |
| Customizing Db2 Cloning Tool for the first time Follow this roadmap if you do not have a customized version of Db2 Cloning Tool, and you need to customize it for the first time. | “Roadmap: Customizing Db2 Cloning Tool for the first time” on page 77 | |

| Task | Link to detailed instructions | Status |
|--|---|--------|
| Customizing a different version of Db2 Cloning Tool Follow this roadmap if you previously customized a version of Db2 Cloning Tool and want to use the same parameter values to customize a different version. | “Roadmap: Customizing a new version of Db2 Cloning Tool from a previous customization” on page 78 | |
| Recustomizing Db2 Cloning Tool Follow this roadmap if you already customized Db2 Cloning Tool but want to change one or more parameter values. | “Roadmap: Recustomizing Db2 Cloning Tool” on page 79 | |

Set up your environment prior to customization

Prior to beginning the customization process, ensure that your environment meets all requirements, that you have installed all prerequisite software, and that you have considered how you want to customize optional features.

Verify that your environment meets software requirements

The operation of Db2 Cloning Tool depends on the following software levels:

- z/OS 2.1 and later.
- Db2 11 or Db2 12, until end of support for the given Db2 version. See the topic “Db2 12 function level support” on [page 9](#) for information about specific Db2 12 function level toleration and support in Db2 Cloning Tool.

Restriction: Db2 table space cloning of archive tables is not supported.

Restriction: Db2 table space cloning using log apply is not supported for PBR RPN table spaces.

- Any available new PTFs. The same level of Db2 Cloning Tool maintenance is required for all cloning jobs.
- One or more of the cloning tools listed in section “[Volume copy products supported by Db2 Cloning Tool](#)” on [page 14](#), or slow copy utilities, such as DFSMSdss copy or Innovation Data Processing products.
- If you will use Db2 Cloning Tool’s stored procedure for subsystem cloning, the Db2 administrative task scheduler must be configured and available on the Db2 systems where the stored procedure will be run.
- If you will use IBM FlashCopy for Db2 Cloning Tool Table Space Cloning, FlashCopy version 2 is required.
- If you plan to use fast replication via the DFSMSdss (ADRDSSU) COPY utility, review the following references prior to performing a cloning. The information in these links might help you avoid cases when a FlashCopy target device may be in conflict with a copy services status:
 - The [Combining copy services operations](#) topic in the z/OS DFSMS Advanced Copy Services documentation.
 - The [Preserve Mirror FlashCopy](#) topic in the z/OS DFSMSdss Storage Administration documentation.
- If you will use the Db2 Cloning Tool ISPF interface, a minimum region size of 50000 KB is required.
- If you will use Db2 Cloning Tool Table Space Cloning, UNIX System Services access is required. The user ID under which the source and the target jobs run must have an OMVS segment in its RACF® profile. To check whether the user ID has an OMVS segment in its profile, use the following command:

```
LU userid OMVS
```

To add an OMVS segment to a user ID's RACF profile, use the following command:

```
ADDUSER ddfuid OMVS(UID(nnn))
```

Verify that Db2 Cloning Tool has been installed successfully

See the Program Directory for IBM Db2 Cloning Tool for z/OS, GI10-8910-00, for installation instructions.

Verify that Tools Customizer for z/OS has been installed successfully

Tools Customizer for z/OS is a component of IBM Tools Base for z/OS (5655-V93), which is available free of charge. Tools Customizer for z/OS provides a standard approach to customizing IBM Db2 for z/OS Tools.

See the Program Directory for IBM Tools Base for z/OS, GI10-8819 for installation instructions.

Verify that your environment meets security requirements

You can prevent unauthorized personnel from executing Db2 Cloning Tool commands. To use Db2 Cloning Tool Table Space Cloning, specific authorizations are also required.

In addition to user authorizations, some functions require other authorizations; refer to [“Function authorization requirements”](#) on page 23.

Authorizations required for Db2 Cloning Tool are:

- Db2 Cloning Tool requires ALTER authority to both the source and target user catalogs. By protecting ALTER authority, unauthorized personnel are prevented from executing the COPY and RENAME commands.
- Db2 Cloning Tool requires UPDATE authority to the target data sets. By protecting UPDATE authority, unauthorized personnel are prevented from executing the DB2UPDATE command.
- The COPY command requires authorization to ADRDSSU ADMIN. By protecting ADRDSSU ADMIN, unauthorized personnel are prevented from executing the COPY command. This only applies for FlashCopy where ADRDSSU is used.
- The COPYCHECK command could be used by an unauthorized person using an authorized person's journal file. To prevent this, secure the journal files.

Authorizations for Db2 Cloning Tool Table Space Cloning are:

- SELECT authority on the Db2 catalog tables for both source and target subsystems
- EXECUTE authority on the CKZPLAN on both source and target subsystems
- DISPLAYDB authority on both source and target databases
- STOPDB authority on the target database
- STARTDB authority on the target database
- ALTER authority for all target tables with identity columns
- If FUZZY-COPY(Y) is used to not stop the source, and ADRDSSU is used, then RACF authority for “TOLERATE(ENQF)” is required
- If FUZZY-COPY(N) (the default) is used, then STOPDB and STARTDB authority for the source database is required

If your site uses ACF2 to restrict TSO command use, you may need to add the TSO commands that Db2 Cloning Tool uses to the ACF2 Command Limiting table. The TSO commands that Db2 Cloning Tool uses are CKZARGSZ, CKZWMMAIN, CKZ2MAIN, CKZ00500, and CKZ00600.

Function authorization requirements

The level of authority required to perform certain Db2 Cloning Tool Subsystem Cloning and Db2 Cloning Tool Table Space Cloning is detailed in this topic.

For Db2 Cloning Tool subsystem or volume cloning

- Db2 Cloning Tool COPY and RENAME commands require ALTER authority to both the source and target user catalogs. Db2 Cloning Tool does NOT update the source catalog but requires ALTER authority to access the catalog via direct access.
 - The source ICF catalog(s) requires ALTER because an IDCAMS VERIFY and an IDCAMS EXAMINE INDEXTEST NODATATEST is done prior to reading it to ensure the source catalog is not broken.
 - The target ICF catalog(s) requires ALTER because it is opened and updated using a routine that manages catalog updates.
- When using DATA-MOVER PGM(ADRDSU), the Db2 Cloning Tool Subsystem Cloning COPY command might require the user ID that is running the command to have specific access to STGADMIN profiles in the FACILITY class, as follows:
 - The ADMINISTRATOR operand is always specified when invoking DFSMSdss. To avoid WTORs, ADRDSU ADMINISTRATOR is used to gain permission to overlay the target volume VTOCIX and/or VVDS during the COPY process. Because the ADMINISTRATOR operand is generated, the user ID running COPY must have READ access to FACILITY class profile STGADMIN.ADR.STGADMIN.COPY.
 - If FCSETGTOK is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCSETGT, if it exists.
 - If FCTOPPRCPPRIMARY is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCTOPPRCP, if it exists.
 - If FCTOXRCPPRIMARY is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCTOXRCP, if it exists.
 - If FlashCopy is being used, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FLASHCPY, if it exists.
 - If CONSISTENT(YES) is specified, READ access is required to FACILITY class profiles STGADMIN.ADR.COPY.FCFREEZE and STGADMIN.ADR.CGCREATE, if they exist.
 - If the CONCURRENT option is being used, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.CNCURRENT, if it exists.
- If Db2 Cloning Tool is going to relabel the target volumes, ICKDSF REFORMAT requires RACF volume access of ALTER to RACF class DASDVOL or READ access to FACILITY class profile STGADMIN.ICK.REFORMAT. Db2 Cloning Tool will relabel the target volumes as part of COPY when VOLPAIRSDEVN or VOLPAIRSDEVN-DDN is used, or as part of VOLOPTIONS when SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP are used.
- Db2 Cloning Tool RENAME command invokes ICKDSF to rebuild the VTOCIX, if VTOCIX_REBUILDER = IBM is specified in CKZINI or the volume is an Extended Address Volume. ICKDSF requires RACF volume access of ALTER to RACF class DASDVOL or READ access to FACILITY class profile STGADMIN.ICK.BUILDIX. If the default, VTOCIX_REBUILDER = MSC, is specified in CKZINI, ICKDSF is not used, unless the volume is an Extended Address Volume for which ICKDSF will always be used to rebuild the VTOCIX.
- Db2 Cloning Tool COPY-BY-DS command requires READ authority to the source data sets and ALTER authority to the target data sets, and might require the user ID that is running the command to have specific access to STGADMIN profiles in the FACILITY class, as follows:
 - If FCTOPPRCPPRIMARY is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCTOPPRCP, if it exists.
 - If FCTOXRCPPRIMARY is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCTOXRCP, if it exists.

- If FlashCopy is being used, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FLASHCPY, if it exists.
- If Db2 Cloning Tool RENAME has NOTRENAMED (DELETE) in effect, READ access to FACILITY class profile STGADMIN.IGG.DLVVRNVR.NOCAT is required.
- Db2 Cloning Tool DB2UPDATE command requires authority to the target data sets.
- Db2 Cloning Tool COPYCHECK command requires UPDATE to the journal file.
- Db2 Cloning Tool FINDUCATS command invokes DCOLLECT to identify ALIAS names of the source volume data sets in order to identify the correct source user catalogs. DCOLLECT controls access to the DCOLLECT function, by issuing a security (RACF) check for a facility class profile of STGADMIN.IDC.DCOLLECT. If this profile exists, then Read authority is necessary.
- Db2 Cloning Tool DB2FIX command requires either Db2 install SYSADM or SYSADM2 authority.
- Db2 Cloning Tool DB2SQL command requires either Db2 install SYSADM or SYSADM2 authority.
- Db2 Cloning Tool DB2SETLOG command requires authorization to connect to the Db2 subsystem and to issue Db2 SET LOG commands.
- Db2 Cloning Tool DB2START command requires authorization to issue z/OS START Db2 commands and to connect to the Db2 subsystem. When the SPECIAL or MAINT keyword is used, it requires either Db2 install SYSADM or SYSADM2.
- Db2 Cloning Tool DB2STOP command requires authorization to connect to the Db2 subsystem and issue the Db2 STOP DB2 command. When the Db2 subsystem is running in maintenance mode, it requires either Db2 install SYSADM or SYSADM2.
- Db2 Cloning Tool RENAME command requires authorization to create an EMCS console and issue z/OS MODIFY CATALOG commands. Update authorization to the VVDS data sets on the target volumes is also required.
- Db2 Cloning Tool DB2UPDATE command (if DB2XCFCLEAN(Y) is specified) requires authority to clean up XCF structures. The ability to clean up XCF structures is controlled with profiles in the FACILITY class with names of IXLSTR . structurename. If profiles exist that cover the target Db2 structures, then UPDATE access is necessary.
- Db2 Cloning Tool DB2XCFCLEAN command requires authority to clean up XCF structures. The ability to clean up XCF structures is controlled with profiles in the FACILITY class with names of IXLSTR . structurename. If profiles exist that cover the target Db2 structures, then UPDATE access is necessary.
- Db2 Cloning Tool DB2GETBACKINFO command requires authority to issue HSM LIST COPYPOOL commands. The ability to issue HSM LIST COPYPOOL commands is controlled with profiles in the FACILITY class with names of STGADMIN.ARC.LC.copypoolname. If profiles exist that cover the copypoolname being used, then READ access is necessary.
- Db2 Cloning Tool RESTORE-FROM-DUMPTAPES command requires the following:
 - READ access to HSM dump tapes (for example, TAPEVOL profile HSMHSM, or READ access by data set name).
 - RACF volume access of ALTER to RACF class DASDVOL.
 - READ access to FACILITY class profile STGADMIN.ICK.REFORMAT.
 - READ access to STGADMIN.ADR.STGADMIN.RESTORE for the target volumes.
- Db2 Cloning Tool BCSCLEAN command requires READ authority to profiles in the FACILITY class with names of STGADMIN.IGG.DIRCAT and STGADMIN.IGG.DELETE.NOSCRTCH.

For Db2 Cloning Tool Table Space Cloning

- SELECT authority on the Db2 catalog tables is required for both source and target subsystems.
- EXECUTE authority on the CKZ plan is required on both source and target subsystems.
- DISPLAYDB authority is required on both source and target databases.
- STOPDB authority is required on the target database.

- STARTDB authority is required on the target database.
- ALTER authority is required for all target tables with identity columns.
- DATAACCESS authority on the target Db2 subsystem is required when cloning objects with XML columns, for updating the sequence numbers that are used for the DB2_GENERATED_DOCID_FOR_XML columns.
- If FUZZY-COPY(Y) is used to not stop the source, and ADRDSSU is used, then RACF authority for “TOLERATE(ENQF)” is required.
- If FUZZY-COPY(N) (the default) is used, then STOPDB and STARTDB authority for the source database is required.
- If the TCP/IP server job is run as a started task, SYSCTL authority must be granted to the started task's user ID on the target Db2 subsystem.
- When using DATA-MOVER PGM(ADRDSSU), the Db2 Cloning Tool Table Space Cloning COPY command might require the user ID that is running the command to have specific access to STGADMIN profiles in the FACILITY class, as follows:
 - If FCTOPPRCPPRIMARY is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCTOPPRCP, if it exists.
 - If FCTOXRCPPRIMARY is specified, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FCTOXRCP, if it exists.
 - If FlashCopy is being used, READ access is required to FACILITY class profile STGADMIN.ADR.COPY.FLASHCPY, if it exists.

Use of the U.S.A. EBCDIC code set

Db2 Cloning Tool uses the U.S.A. EBCDIC code set for specification and display of EBCDIC characters and for the extended ACS masking characters used for filtering.

If the code tables used by your installation are different, then you need to enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals.

- Data supplied as input to batch programs or input to ISPF panels:

For product code shipped in binary, when specifying input where the product takes special action based on specific characters, you must enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals, according to the U.S.A. EBCDIC code set.

For example, if an exclamation mark (!) is called for, and your code tables do not translate the ! character to a hexadecimal 5A, you must enter the character that your code table will translate to a 5A.

- Distributed ISPF panels:

Do not change distributed ISPF panels. Program code may reference ISPF panel attribute bytes. A panel change that affects an attribute byte may cause a program error.

- Product output:

Depictions of product output shown in the product manuals are based on the U.S.A. EBCDIC code set. Actual output may vary if your EBCDIC code tables are different.

- Extended ACS masking characters:

Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set. For more information about extended ACS masking characters, see [“Filtering pattern masks”](#) on page 26.

Filtering pattern masks

Many commands allow specification of powerful filtering masks. These filtering pattern masks can be specified with extended ACS masking characters, as specified in the table.

Note: Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set.

| Mask | Hex | Description |
|------|-----|--|
| * | 5C | <p>A single asterisk (*) represents 0–<i>n</i> characters. It can be used before and/or after an item to designate a wildcard character position. You cannot use an asterisk in the middle portion of a single item (except for data set names). Use the percent sign (%) character for that purpose.</p> <p>Example: *PROD</p> <p>In this example, an item is selected if the last four characters are PROD regardless of the starting characters. Using a single asterisk character before and after an item (i.e. *ZREM*) means that item is to be selected if any character string matches ZREM in its name.</p> |
| % | 6C | <p>A percent sign (%) denotes a single-character placeholder value, which can be alpha-numeric or any special character. The percent sign can be used in any position order.</p> <p>Example: CRM%%ER6</p> <p>In this example, an item is selected if it is eight characters in length, the first three characters are CRM, and the last three characters are ER6. The two middle placeholder values (%%) can be any characters.</p> |
| < | 4C | <p>Similar to the percent sign (%) value, the "less than" character (<) denotes a placeholder value <i>for only alpha characters</i>. The < character can be used in any position order.</p> <p>Example: CR<<ER*</p> <p>In this example, an item is selected if the first two characters are CR, the third and fourth characters are alpha, and the fifth and sixth characters are ER. Any remaining characters are also allowed.</p> |

Table 2. Filtering pattern masks (continued)

| Mask | Hex | Description |
|------|------|---|
| > | 6E | <p>Similar to the percent sign (%) value, the "greater than" character (>) denotes a placeholder value <i>for only numeric characters</i>. The > character can be used in any position order.</p> <p>Example:</p> <p>CR>>ER*</p> <p>In this example, an item is selected if the first two characters are CR, the third and fourth characters are numeric, and the fifth and sixth characters are ER. Any remaining characters are also allowed.</p> |
| ** | 5C5C | <p>Using two asterisks (**) allows compatibility with standard ACS and <i>DFSMSdss</i> filtering masks. Only used for <i>partially</i> qualified data set names. The * or ** characters can be used in any qualifier position to denote a <i>wildcard</i> node.</p> <p>Example:</p> <p>CRFM*.*VER.**</p> <p>In this example, a data set entry is selected if it has at least two qualifiers, the first qualifier starts with CRFM, and the second qualifier ends with VER. Any remaining qualifiers are also allowed.</p> <p>Note: As with common data set name masking, any combination of *, **, %, < and/or > characters can be used for the item mask value.</p> |
| ! | 5A | <p>Similar to the percent sign (%) value, the exclamation point character (!) denotes a placeholder value <i>for only national characters</i>, based on the U.S.A. EBCDIC code set. Refer to the topic “Use of the U.S.A. EBCDIC code set” on page 25 for more information. The exclamation point (!) can be used in any position order.</p> <p>National characters (based on the U.S.A. EBCDIC code set) are: @ # \$</p> |

Worksheets: Gathering required data set names

Identify and record the data set names that will be used during the customization process and make sure that requirements for certain data sets are met.

Data set names for Tools Customizer

Identify and record the following Tools Customizer data set names:

| Data set name | Description | Special requirements | Your data set name |
|---------------|---------------------------------------|----------------------|--------------------|
| SCCQDENU | Metadata library for Tools Customizer | | |

| Data set name | Description | Special requirements | Your data set name |
|---------------|---|--|--------------------|
| SCCQLOAD | Executable load module library for Tools Customizer | | |
| SCCQMENU | ISPF messages for Tools Customizer | | |
| SCCQPENU | ISPF panels for Tools Customizer | | |
| SCCQSAMP | Sample members for Tools Customizer | | |
| SCCQTENU | Table library for Tools Customizer | You must have write access to this data set. | |

Data set names of Db2 Cloning Tool

Identify and record the following Db2 Cloning Tool data set names. During the customization process, you will enter the following values on panel CCQPPRD.

| Data set name | Description | Special requirements | Your data set name |
|---|---|---|--------------------|
| Installation high-level qualifier | The high level qualifier of the Db2 Cloning Tool installed libraries. | | |
| Installation high-level qualifier of previous release of Db2 Cloning Tool | The high level qualifier of the previous release of Db2 Cloning Tool installed libraries. | Only needed if upgrading from a previous release. | |
| SCKZDBRM | DBRM library for Db2 Cloning Tool | | |
| SCKZDENU | Metadata library for Db2 Cloning Tool | | |
| SCKZJCL | Sample members for Db2 Cloning Tool | | |
| SCKZLOAD | Executable load module library for Db2 Cloning Tool | You must APF authorize this data set. | |
| SCKZMENU | ISPF messages for Db2 Cloning Tool | | |
| SCKZPARM | Parameter library for Db2 Cloning Tool | | |
| SCKZPENU | ISPF panels for Db2 Cloning Tool | | |
| SCKZSLIB | Skeleton library for Db2 Cloning Tool | | |

Data set names of other libraries

Identify and record the following data set names. During the customization process, you will enter the following values on the Setup panel.

| Data set name | Description | Special requirements | Your data set name |
|--------------------------|--|--|--------------------|
| Discover output data set | <p>Contains the output that is generated when you run the Db2 Cloning Tool Discover EXEC.</p> <p>The Db2 Cloning ToolDiscover EXEC retrieves the metadata and values for the parameters from a previous customization of Db2 Cloning Tool.</p> <p>The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the Tools Customizer Settings panel or the Discover Customized Product Information panel.</p> | You must have write access to this data set. | |
| Data store data set | <p>Contains product, LPAR, and Db2 parameter values, and Db2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, Db2 subsystems, and LPAR when you customize products on the local LPAR.</p> <p>The default name of the data set is DB2TOOL.CCQ110.DATASTOR. You can change the default value on the Tool Customizer Settings panel.</p> | You must have write access to this data set. | |

| Data set name | Description | Special requirements | Your data set name |
|-------------------------------|--|--|--------------------|
| Product customization Library | <p>Contains the customization jobs that Tools Customizer generates for Db2 Cloning Tool.</p> <p>To customize Db2 Cloning Tool, submit the members of the data set in the order in which they are displayed on the Finish Product Customization panel. The data set naming convention is:</p> <pre data-bbox="500 646 787 682"><i>hlq.\$LPAR-name\$.xyzvrm</i></pre> <p>where:</p> <ul style="list-style-type: none"> • <i>hlq</i> is the value of the Customization library qualifier field on the Tools Customizer Settings panel (CCQPSET) • <i>LPAR-name</i> is the four-character LPAR name • <i>xyzvrm</i> is the three-letter product identifier with the version, release, and modification level <p>For example, the data set name might be DB2TOOL.PRODUCT.CUST . \$MVS1\$.XYZ410.</p> | You must have write access to this data set. | |

APF authorizing load libraries

Some of the programs in Db2 Cloning Tool load libraries must be APF-authorized to run.

About this task

The SCKZLOAD library for Db2 Cloning Tool must have APF authorization. Consult your Systems Programmer to have this library added into the APF list and ensure appropriate access controls have been established

Procedure

- Include the SCKZLOAD load library as part of your authorized list.

Note: In general, do not place product modules in LINKLST because doing so could introduce a member name conflict with another product.

Worksheets: Gathering parameter values for Tools Customizer

During the customization process, you must provide parameter values for Db2 Cloning Tool, for Db2, and for your LPAR.

Use the worksheets in this topic to record the appropriate parameter settings for your purposes, and then use these worksheets during the customization process. The worksheets are organized based on the order of the customization panels in the Tools Customizer.

After the customization jobs are generated, they are grouped by the job sequence number. In this topic, the jobs use an 8-character member naming convention that follows the format of *ssjjjdd*, where:

- *ss* is the job sequence number, which is an alphabetic character (A - Z) followed by a numeric character (0 - 9). For example, a job sequence number is A0, A1, ..., Z9.
- *jjjj* is derived from the job template name. The product assigns the job template name.
- *dd* is a set of two alphanumeric characters (AA - 99) that Tools Customizer assigns to identify a Db2 entry.

For example, the job shown as *ssCF2Add* in this topic might be generated as A4CF2AAA.

Customization values for the Discover EXEC

Description

Use the following worksheet to identify and record the customization values for the Tools Customizer Discover EXEC. The values in this worksheet are for extracting information from a product that has already been customized. During the customization process, you will enter these values on panel CCQPDSC.

Note: Complete this worksheet only if you are recustomizing a product that has previously been customized by using Tools Customizer.

| Table 3. Customization values for the Discover EXEC | | |
|---|--|------------|
| Parameter | Sample or default value | Your value |
| Discover EXEC for Extracting Information from an Already Customized product | | |
| Discover EXEC library The fully qualified data set name that contains the product Discover EXEC. | The metadata library that you specified on the Specify the Metadata Library panel (CCQPHLQ). | |
| Discover EXEC name The name of the Discover EXEC. | CKZDISC | |
| Discover output data set The name of the data set for the output from the product Discover EXEC. | The name of the discover output library that you entered on the settings panel. | |
| Information for Discover EXEC section | | |
| Previous release Db2 Cloning Tool CKZINI library The previous release of the Db2 Cloning Tool CKZINI library data set name. | DB2TOOL.CKZ310.SCKZPARM | |

Table 3. Customization values for the Discover EXEC (continued)

| Parameter | Sample or default value | Your value |
|--|-------------------------|------------|
| <p>Db2 Cloning Tool ISPF control data set</p> <p>The data set name of the Db2 Cloning Tool ISPF interface control data set. Leave this field blank if you do not want DISCOVER to retrieve Db2 information from a previous control data set. If you have installed other IBM Db2 tools that require this control data set, you can specify the same control file data set name that you used for the other Db2 tools.</p> | blank | |

Product to Customize section

Description

The parameters that are listed in the Product to Customize section are read-only. They contain information that was provided on other panels, by Tools Customizer, or by the Db2 Cloning Tool metadata data set.

Table 4. Product to customize section

| Parameter | Discovered? | Source of this value |
|--|-------------|--|
| <p>Product metadata library</p> <p>This value is the library that you specified on the Specify the Product to Customize panel. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.</p> | No | This value is specified on the Specify the Product to Customize panel (CCQPHLQ) |
| <p>LPAR</p> <p>The LPAR field displays the LPAR on which you are customizing Db2 Cloning Tool.</p> | No | This value is supplied by Tools Customizer. |
| <p>Product name</p> <p>This value displays the product that is being customized. In this example, IBM Db2 Cloning Tool should be displayed in this field. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.</p> | No | This value is provided by the product metadata file. |
| <p>Version</p> <p>The Version field displays the version, release and maintenance of the product you are customizing in the format Vn.Rn.nn.</p> | No | This value is provided by the product metadata file. |

Common parameters section

Description

The parameters in this section are required for all customizations. During the customization process, you will enter these values on panel CCQPPRD.

| <i>Table 5. Common parameters section</i> | | | | |
|--|-----------|-------------|----------------|------------|
| Parameter | Required? | Discovered? | Default value | Your value |
| HLQ of current release The high-level qualifier of the current release of the Db2 Cloning Tool libraries. | Yes | Yes | DB2TOOL.CKZ320 | |
| Db2 plan name The Db2 plan name to use for Db2 Cloning Tool. This plan name is used for both Db2 Cloning Tool Subsystem Cloning and Db2 Cloning Tool Table Space Cloning plans. This parameter corresponds to the CKZINI DB2_PLAN parameter. | Yes | Yes | CKZPLAN | |

Task: Create and update CKZINI

Description

This task builds the job to run INIMERGE, which merges a previous CKZINI member or creates a new CKZINI member, and the job to update the CKZINI member with new values provided in this release. The CKZINI member is the product's initialization member, and defines global information regarding Db2 Cloning Tool usage and options within your installation. Two jobs will be generated from this step. During the customization process, you will enter these values on panel CCQPPRD.

This task is *required*.

Jobs generated

This task generates the following jobs using the listed templates:

| <i>Table 6. Generated jobs for running INIMERGE</i> | | | |
|---|----------|--|--|
| Jobname | Template | Description | Required authority to run the job |
| ssIMR | CKZIMR | Run INIMERGE to merge a previous CKZINI or create a new CKZINI member. | The user ID that runs the ssIMR job must have the following access to Db2 Cloning Tool data sets: <ul style="list-style-type: none"> • Read access to the previous release SCKZPARM data set • Update access to the current release SCKZPARM data set • Execute access to the current release SCKZLOAD data set |

Table 6. Generated jobs for running INIMERGE (continued)

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|-----------------------------|---|
| ssIUP | CKZIUP | Run update of CKZINI member | <p>The user ID that runs the ssIUP job must have the following access to Db2 Cloning Tool data sets:</p> <ul style="list-style-type: none"> • Update access to the current release SCKZPARM data set • Read access to the current release SCKZDENU data set |

Required authority

Refer to the job list for the required authorizations.

Table 7. Run INIMERGE parameters

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|------------------|------------|
| <p>Run INIMERGE This step builds the job to run INIMERGE.</p> | Yes | No | Selected | |
| <p>Location of the CKZINI currently in use The location of the CKZINI member that is currently in use. If the CKZINI that is currently in use is in the library for the current release, specify CURRENT-RELEASE. If the location of the CKZINI currently in use is in the previous release, specify PREVIOUS-RELEASE. If a CKZINI is not currently in use, specify NONE.</p> | Yes | No | PREVIOUS-RELEASE | |
| <p>HLQ of previous release The high-level qualifier of the previous release of the Db2 Cloning Tool libraries. This value is ignored if this is a new customization of Db2 Cloning Tool.</p> | Yes | Yes | DB2TOOL.CKZ310 | |
| <p>Run CKZINI update This step builds the job to update the CKZINI member.</p> | No | No | Not selected | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| <p>Space management Identifies the space management product being used on the system. This parameter specifies the volume serial that should be used as an indication that a data set has been migrated. HSM and ABR(#) use a volume serial of MIGRAT. DMS uses a volume serial of ARCIVE. Due to an ISPF limitation, you must specify ABR{#} instead of ABR(#). This parameter corresponds to the CKZINI SPACE_MANAGEMENT parameter.</p> | No | Yes | HSM | |
| <p>Concurrent executions Indicates if there can be multiple jobs that can run concurrently and need exclusive control over the same BSC. N indicates that the Db2 Cloning Tool job will fail if another job has exclusive control over the BCS. Y indicates that the Db2 Cloning Tool job will wait if another job has exclusive control over the BCS. This parameter corresponds to the CKZINI CONCURRENT_EXECUTIONS parameter.</p> | No | Yes | N | |
| <p>Concurrent executions wait time If Concurrent executions is set to Y, specifies the maximum wait time in minutes. The maximum value that can be specified is 999. This parameter corresponds to the CKZINI CONCURRENT_EXECUTIONS_WAIT_TIME parameter.</p> | Yes | Yes | 5 | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---|------------|
| <p>Catalog work data set attributes Specifies the attributes to be used for the catalog work data sets. These data sets contain catalog entries captured during the COPY step. This parameter controls the attributes for these data sets if they are not specified in the COPY command. Specify attributes in TSO ALLOCATE syntax; for example: UNIT(SYSALLDA) SPACE(10 10) CYLINDERS. This parameter corresponds to the CKZINI CATWORK-ATTR parameter.</p> | Yes | Yes | UNIT(SYSALLDA) SPACE(10 10) CYLINDERS | |
| <p>Target volumes should be empty Specifies if the target volumes should be empty to be used as targets of the COPY process. N indicates that a target volume that is not empty can be used as a target of the COPY process. Y indicates that a target volume that is not empty cannot be used as a target of the COPY process. This parameter corresponds to the CKZINI TARGET_VOLS_SHOULD_BE_EMPTY parameter.</p> | No | Yes | N | |
| <p>GDG all migrated Specifies the action for RENAME to take for a GDG when all GDS are migrated. SKIP indicates that the BCS update should be skipped. RETAIN indicates that the migrated entries should be copied as is to the target BCS. This parameter corresponds to the CKZINI GDG_ALL_MIGRATED parameter.</p> | No | Yes | SKIP | |
| <p>GDG all migrated RC If GDG all migrated is set to RETAIN, specifies the return code to use. This parameter corresponds to the CKZINI GDG_ALL_MIGRATED_RETAIN_RC parameter.</p> | No | Yes | 4 | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| <p>GDG empty Specifies the action for RENAME to take for a GDG that is empty. SKIP indicates that the BCS update should be skipped. RETAIN indicates that the empty GDG should be added to the target BCS. This parameter corresponds to the CKZINI GDG_EMPTY parameter.</p> | No | Yes | SKIP | |
| <p>GDG empty RC If GDG empty is set to RETAIN, specifies the return code to use. This parameter corresponds to the CKZINI GDG_EMPTY_RETAIN_RC parameter.</p> | No | Yes | 4 | |
| <p>GDG migrated Specifies the action for RENAME to take for a GDG that has at least one GDS on a cloned volume and has one or more GDS are migrated. ERROR indicates that the BCS update should terminate. RETAIN indicates that the GDG should be copied as is to the target BCS. This parameter corresponds to the CKZINI GDG_MIGRATED parameter.</p> | No | Yes | ERROR | |
| <p>GDG migrated RC If GDG migrated is set to RETAIN, specifies the return code to use. This parameter corresponds to the CKZINI GDG_MIGRATED_RETAIN_RC parameter.</p> | No | Yes | 4 | |
| <p>GDG tape Specifies the action for RENAME to take for a GDG that has at least one GDS on a cloned volume and has one or more GDS on tape. ERROR indicates that the BCS update should terminate. RETAIN indicates that the GDG should be copied as is to the target BCS. This parameter corresponds to the CKZINI GDG_TAPE parameter.</p> | No | Yes | ERROR | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| GDG tape RC If GDG tape is set to RETAIN, specifies the return code to use. This parameter corresponds to the CKZINI GDG_TAPE_RETAIN_RC parameter. | No | Yes | 4 | |
| Issue CKZ14141I Specifies when the CKZ14141I message should be issued. ALL indicates the message should be issued for all matching data sets. DASD indicates the message should be issued only for DASD data sets. MIG indicates the message should be issued only for migrated data sets. TAPE indicates the message should be issued only for tape data sets. NOMSG indicates the message should not be issued. This parameter corresponds to the CKZINI ISSUE_CKZ14141I parameter. | No | Yes | ALL | |
| Maximum RENAME tasks Specifies the maximum number of tasks that will be used by the RENAME command if the MAXTASKS keyword is not specified in the RENAME command. This parameter corresponds to the CKZINI MAX_RENAME_TASKS parameter. | Yes | Yes | 5 | |
| Missing usercat disposition Specifies the disposition for target volume data sets where the VVDS catalog back-pointer is not a catalog in the list that is supplied to the COPY step. This parameter corresponds to the CKZINI MISSING_USERCAT_DISP parameter. | No | Yes | KEEP | |
| Missing usercat RC Specifies the return code to be generated for the RENAME command if one or more target volume data sets contain a VVDS catalog back-pointer that is not in the list supplied to the COPY step. This parameter corresponds to the CKZINI MISSING_USERCAT_RC parameter. | No | Yes | 4 | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| Not renamed disposition Specifies the disposition for data sets that do not match a rename mask. This parameter corresponds to the CKZINI NOT_RENAMED_DISP parameter. | No | Yes | KEEP | |
| Not renamed RC Specifies the return code to be generated for the RENAME command if one or more data sets are not renamed. This parameter corresponds to the CKZINI NOT_RENAMED_RC parameter. | No | Yes | 8 | |
| Orphan catalog entry disposition Specifies the disposition of target volume data set catalog entries where in some circumstances the data set is not found on the volume. This parameter corresponds to the CKZINI ORPHAN_CATENTRY_DISP parameter. | No | Yes | KEEP | |
| Orphan catalog entry RC Specifies the return code to be generated for the RENAME command where in some circumstances the data set is not found on the volume. This parameter corresponds to the CKZINI ORPHAN_CATENTRY_RC parameter. | No | Yes | 8 | |
| Recatalog over existing entry Specifies if an existing target ICF catalog entry may be replaced without considering it an error. If N is specified, and a target catalog entry for a renamed data set exists, the process fails. If Y is specified, and a target catalog entry for a renamed data set exists, it will be overlaid. This parameter corresponds to the CKZINI RECATALOG parameter. | No | Yes | N | |
| Temporary data set disposition Specifies the disposition of temporary data sets on the target volumes. This parameter corresponds to the CKZINI TEMP_DATASET_DISP parameter. | No | Yes | DELETE | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| <p>Temporary data set RC Specifies the return code to be generated for the RENAME command if one or more temporary data sets are found on the target volumes. This parameter corresponds to the CKZINI TEMP_DATASET_RC parameter.</p> | No | Yes | 4 | |
| <p>VTOCIX rebuilder Specifies the method for rebuilding the VTOCIX during the RENAME command. For IBM, ICKDSF will be used to rebuild the VTOCIX. For MSC, the on-board VTOCIX rebuild will be used. This parameter corresponds to the CKZINI VTOCIX_REBUILDER parameter.</p> | No | Yes | MSC | |
| <p>Rename error processing Specifies how processing proceeds when a RENAME error is encountered. ABORT will terminate with an RC=8 after the first error to preserve integrity. CONTINUE will continue processing after most errors and the RENAME command will complete with the specified return code unless an error not handled by the CONTINUE logic is encountered.</p> <p> CAUTION: The use of CONTINUE can cause inconsistencies between the contents of the volumes and catalogs.</p> <p>This parameter corresponds to the CKZINI RENAME_ERROR parameter.</p> | No | Yes | ABORT | |
| <p>Rename error RC If Rename error processing is set to CONTINUE, specifies the return code to be used. This parameter corresponds to the CKZINI RENAME_ERROR_CONTINUE_RC parameter.</p> | No | Yes | 4 | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| <p>Issue VCLOSE Specifies whether a catalog modify VCLOSE command will be issued as part of the volume RENAME processing. NO specifies that the modify command will not be issued. BEFORE specifies that the modify command will be issued only before the VVDS is updated. AFTER specifies that the modify command will be issued only after the VVDS has been updated. YES specifies that the modify command will be issued both before and after the VVDS has been updated. This parameter corresponds to the CKZINI ISSUE_VCLOSE parameter.</p> | No | Yes | YES | |
| <p>Issue VCLOSE scope If Issue VCLOSE is set to YES, BEFORE, or AFTER, this field specifies the scope of the VCLOSE command. LOCAL specifies that the VCLOSE command will be issued only on the system that RENAME is running on. SYSPLEX specifies that the VCLOSE command will be issued on the local system, and the command will be routed to all the other systems in the sysplex, via an MVS ROUTE *OTHER command after the VVDS has been updated. This parameter corresponds to the CKZINI ISSUE_VCLOSE_SCOPE parameter.</p> | No | Yes | LOCAL | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| <p>Sysplex group name all This parameter specifies the scope of the ROUTE command execution. *ALL indicates that the ROUTE command will be run on all systems of the sysplex. Entering a system group name in this field indicates that the ROUTE command will be run on systems that are defined in that system group name. This parameter corresponds to the CKZINI parameter SYSPLEX_GROUPNAME_ALL. This value is required. Valid values are *ALL and the name of a system group name. The default value is *ALL.</p> | No | Yes | *ALL | |
| <p>DB2 XCF CLEAN Specifies that DB2UPDATE should clean up the target Db2 data sharing group XCF structures and group members. Y specifies that the XCF structures and group members should be cleaned up. N specifies that the XCF structures and group members should not be cleaned up. This parameter corresponds to the CKZINI DB2_XCF CLEAN parameter.</p> | No | Yes | Y | |
| <p>Db2 plan name The Db2 plan name to use for Db2 Cloning Tool. This plan name is used for both subsystem cloning and table space cloning plans. This parameter corresponds to the CKZINI DB2_PLAN parameter.</p> | No | Yes | CKZPLAN | |
| <p>MIM GDIF If you have CA-MIM/MII with multiple systems and shared DASD, this field should be set to Y to ensure that when CA-MIM/MII GDIF is inactive, the Db2 Cloning Tool data sets are protected from data corruption. This parameter corresponds to the CKZINI MIM_GDIF parameter.</p> | No | Yes | N | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Max COPY RC Specifies the maximum return code for data set copy. When the specified return code is exceeded, the job ends in error. This parameter only applies to copy processing return codes. This allows one or more copies to fail and the others to continue. This parameter corresponds to the CKZINI MAX_COPY_RC parameter. | No | Yes | 0 | |
| Max RC Specifies the maximum job return code. When the specified return code is exceeded, the job ends in error. This parameter corresponds to the CKZINI MAX_RC parameter. | No | Yes | 0 | |
| Maximum subtasks Specifies the number of subtasks to start. This parameter can be used to reduce the elapsed time of long-running source and target jobs by allowing multiple I/O operations to run concurrently. This parameter corresponds to the CKZINI MAX_SUBTASKS parameter. | Yes | Yes | 1 | |
| TCP/IP server port number Specifies the port that the TCP/IP server is to use to wait for requests from the source job. This parameter corresponds to the CKZINI TCPIP_SERVER_PORT parameter. | No | Yes | 5099 | |
| TCP/IP started task name Specifies the name of the TCP/IP address space that the TCP/IP server and source job connect to. This parameter corresponds to the INI TCPIP_STC_NAME parameter. | Yes | Yes | TCPIP | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| <p>TCP/IP encryption enable Specifies if the TCP/IP communications between the source job and the TCP/IP server job, and the target job and the source TCP/IP server job, will be encrypted. Y specifies that TCP/IP communications is encrypted. N specifies that TCP/IP communications is not encrypted. This parameter corresponds to the CKZINI TCP/IP_ENCRYPTION_ENABLE parameter.</p> | Yes | Yes | No | |
| <p>TCP/IP key label Specifies the key label for the TCP/IP communications encryption between the source job and the TCP/IP server job, and the target job and the source TCP/IP server job. This parameter corresponds to the CKZINI TCP/IP_KEY_LABEL parameter. This value is optional. The default value is empty.</p> | No | Yes | blank | |
| <p>Always copy index spaces Specifies how index spaces are included in LISTDEFS. If Y is specified, all index spaces are included for every table space included in a LISTDEF. No INCLUDE INDEXSPACES syntax is required in the LISTDEF. This parameter corresponds to the CKZINI ALWAYS_COPY_INDEXSPACES parameter.</p> | No | Yes | N | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| <p>Auto start source spaces Specifies if the source job will start each source Db2 table space and index space after the copy process is complete. Y specifies that starts of the source table and index spaces are to be done after the copy is complete. N specifies that the source spaces are left stopped after the copy is complete. R specifies that the source table and index spaces are restored to the status they were before executing the source job. This parameter corresponds to the CKZINI AUTO_START_SOURCE_SPACE parameter.</p> | No | Yes | Y | |
| <p>Auto start target spaces Specifies if SYNCDB2 processing in the target job issues a Db2 START DATABASE SPACENAM command for the table space or index space being processed after all of the sync I/O to the data set is complete. This parameter corresponds to the CKZINI AUTO_START_TARGET_SPACE parameter.</p> | No | Yes | Y | |
| <p>Auto stop target spaces Specifies if the source job should stop the target Db2 table spaces. Y specifies that stops of the target table spaces and index spaces be done before the copy. N specifies that stops of the target table spaces and index spaces will not be done. This parameter corresponds to the CKZINI AUTO_STOP_TARGET_SPACE parameter.</p> | No | Yes | Y | |
| <p>Copy if no target object Specifies if source table spaces or index spaces should be copied if no target table spaces or index spaces exist. This parameter corresponds to the CKZINI COPY_IF_NO_DB2_TARGET_OBJECTS parameter.</p> | No | Yes | N | |

Table 7. Run INIMERGE parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Number of data sets per DSS copy cmd Specifies the number of data sets to send to DSS in a single copy command. This parameter corresponds to the CKZINI DSNS_PER_COPY parameter. | No | Yes | 255 | |
| Number of copy cmds per DSS invocation Specifies the number of DSS copy commands to send to DSS in a single invocation. This parameter corresponds to the CKZINI DSS_COPY_COMMANDS parameter. | No | Yes | 24 | |
| Reset LOGRBA Specifies if the LOGRBA in the target table space and index space data sets should be reset. This parameter corresponds to the CKZINI RESET_LOGRBA parameter. | No | Yes | Y | |
| Pass null storage class to ACS routines If set to Y, a null storage class will be passed to the ACS routines. This allows the ACS storage class routines to assign storage classes to target data sets. If set to N, the input to the ACS will be the source data set's storage class. This parameter corresponds to the CKZINI NULLSTORCLAS parameter. | Yes | Yes | Y | |

Task: Free Db2 packages and plan

Description

This task frees the Db2 packages and plan used by Db2 Cloning Tool Subsystem Cloning, Db2 Cloning Tool Table Space Cloning, and the ISPF interface before a rebind. This task can be used to free old package versions.

One job will be generated from this step. During the customization process, you will enter these values on panel CCQPPRD.

This task is *optional*.

Jobs generated

This task generates the following job using the listed templates:

Table 8. Generated jobs for freeing Db2 packages and plans

| Jobname | Template | Description | Required authority to run the job |
|----------|----------|---|--|
| ssFREEdd | CKZFREE | Free Db2 Cloning Tool packages and plan | FREE authority on the Db2 system and read access to SCKZDBRM |

Required authority

Refer to the job list for the required authorizations.

Table 9. Parameters to free the packages and plans

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Free Db2 packages and plan Creates a job to free the product packages and plan before rerunning the product BIND jobs. One job per Db2 entry will be generated. | No | Yes | Not selected | |

Task: Db2 plan bind

Description

This task binds the Db2 plan used by both Db2 Cloning Tool Subsystem Cloning and Db2 Cloning Tool Table Space Cloning. The plan name (specified in the Common Parameters section) must be the same plan name specified by the parameter DB2_PLAN in the CKZINI member, and must also be the same plan name used in the bind.

For Db2 Cloning Tool Subsystem Cloning, to update the Db2 catalog on the target Db2 subsystem, the DB2SQL and DB2SCHEMA-UPDATE commands have a plan and package that must be installed. If you bind the plan and package on the source Db2 subsystem before cloning, the plan and package will be available for use on the target Db2 subsystem when needed without having to explicitly bind them on the target Db2 subsystem after cloning. The package name can be any value.

Db2 Cloning Tool Table Space Cloning uses static and dynamic SQL to access the catalog and issue Db2 ALTER commands. If you are planning on using Db2 Cloning Tool Table Space Cloning, a bind is required on each subsystem on which table space cloning will execute. This includes all the table space cloning execution types: source, target, source, and TCP/IP server.

One job will be generated from this step. During the customization process, you will enter these values on panel CCQPPRD.

This task is *required*.

Jobs generated

This task generates the following jobs using the listed templates:

Table 10. Generated jobs for Db2 plan binds

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|---|--|
| ssBPTdd | CKZBPT | Db2 Cloning Tool Subsystem Cloning and Db2 Cloning Tool Table Space Cloning plan bind | BIND authority on the Db2 system and read access to SCKZDBRM |

Required authority

Refer to the job list for the required authorizations.

Table 11. Plan bind parameters

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Bind the Db2 Cloning Tool plan Performs a Db2 BIND for the plan used by subsystem cloning and table space cloning. One job per Db2 entry will be generated. | Yes | Yes | Selected | |

Task: Subsystem Cloning tasks

Description

This set of tasks binds the Db2 Cloning Tool Subsystem Cloning packages and defines the Db2 Cloning Tool Subsystem Cloning stored procedure.

To use the Db2 Cloning Tool Subsystem Cloning stored procedure, it must be defined to Db2 and its package must be bound. The package path value used to define the stored procedure should be the same as the package name used to bind the stored procedure. All Db2 Cloning Tool packages are bound using package versions.

This task creates the objects that are required to run the Db2 Cloning Tool Subsystem Cloning stored procedure on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation. During the customization process, you will enter these values on panel CCQPPRD.

Some of these tasks are *required* and some are *optional*.

Jobs generated

This task generates the following jobs using the listed templates:

Table 12. Generated jobs for subsystem cloning tasks

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|---|---|
| ssBSSdd | CKZBSS | Subsystem cloning package bind | BIND authority on the Db2 system and read access to SCKZDBRM |
| ssIDXdd | CKZIDX | Creates optional indexes on the Db2 catalog tables | Authority to create indexes on the Db2 catalog. |
| ssRSPdd | CKZRSP | Drop existing subsystem cloning stored procedure | DROP authority on the stored procedure on the Db2 system |
| ssDSPdd | CKZDSP | Define subsystem cloning stored procedure | CREATE PROCEDURE authority on the Db2 system |
| ssBSPdd | CKZBSP | Bind the subsystem cloning stored procedure package | BIND authority on the Db2 system, and read access to SCKZDBRM |
| ssRSTdd | CKZRST | Drop subsystem cloning global temporary table | DROP TABLE authority for the subsystem cloning global temporary table on the Db2 system |

Table 12. Generated jobs for subsystem cloning tasks (continued)

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|---|---|
| ssDSTdd | CKZDST | Define subsystem cloning global temporary table | CREATE GLOBAL TEMPORARY TABLE authority on the Db2 system |
| ssSP1 | CKZSP1 | Allocate the subsystem cloning stored procedure parameters data set | Authority to create the data set |
| ssSP2 | CKZSP2 | Allocate the subsystem cloning stored procedure cloning data sets | Authority to create the data sets |
| ssSP3 | CKZSP3 | Allocate the stored procedure EXEC library | Authority to create the data set |
| ssSP4 | CKZSP4 | Create the EXEC that is used to invoke the stored procedure | Authority to update the data set |
| ssSP5dd | CKZSP5 | Create the job to invoke the stored procedure | Authority to update the data set |
| ssSPP | CKZSPP | Create the subsystem cloning stored procedure product parameters | Authority to update the data set |
| ssSPSdd | CKZSPS | Create the subsystem cloning stored procedure Db2 system parameters | Authority to update the data set |

Required authority

Refer to the job list for the required authorizations.

Table 13. Plan bind, plan name, and other parameters

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Run Subsystem Cloning package bind Binds the subsystem cloning package. One job per Db2 entry will be generated. | Yes | No | Selected | |
| Create indexes on Db2 catalog Creates optional indexes on the Db2 catalog tables to help improve the performance of the DB2SCHEMA-UPDATE command. | No | No | Not selected | |
| Drop Subsystem Cloning stored procedure Drops the subsystem cloning stored procedure. One job per Db2 entry will be generated. Use this step to drop a previously created stored procedure. | No | No | Not selected | |

Table 13. Plan bind, plan name, and other parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|----------------|------------|
| Define Subsystem Cloning stored procedure Defines the subsystem cloning stored procedure. One job per Db2 entry will be generated. You must run the subsystem cloning stored procedure package bind before you can use the subsystem cloning stored procedure. | No | No | Not selected | |
| Run Subsystem Cloning stored procedure package bind Binds the subsystem cloning stored procedure package. One job per Db2 entry will be generated. You must run the subsystem cloning stored procedure package bind before you can use the subsystem cloning stored procedure. | No | No | Not selected | |
| Drop Subsystem Cloning stored procedure global temporary table Drops the subsystem cloning stored procedure global temporary table. The global temporary table is used by the stored procedure to pass back a result set. One job per Db2 entry will be generated. | No | No | Selected | |
| Define Subsystem Cloning stored procedure global temporary table Defines the subsystem cloning stored procedure global temporary table. The global temporary table is used by the stored procedure to pass back a result set. One job per Db2 entry will be generated. | No | No | Selected | |
| Allocate Subsystem Cloning stored procedure parameters data set Allocates the subsystem cloning stored procedure parameters data set. One job will be generated. The prefix.PARMS data set is created. This data set is used to hold the parameter files used by the stored procedure. | No | No | Not selected | |
| Data set prefix The prefix for the subsystem cloning stored procedure parameters data set. | No | No | DB2TOOL.CKZ.SP | |

Table 13. Plan bind, plan name, and other parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|--------------------------|------------|
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Data set VOLSER The volume serial number (VOLSER) for the subsystem cloning stored procedure parameters data set. If the data set will be managed by SMS, leave this field blank. | No | No | blank | |
| Data set type The data set type for the subsystem cloning stored procedure parameters data set. For a PDS, specify PDS. For a PDSE, specify LIBRARY. | No | No | LIBRARY | |
| Allocate Subsystem Cloning stored procedure cloning data sets Allocates the subsystem cloning stored procedure cloning data sets. One job will be generated. The prefix.JCL data set and prefix.STATUS data set are created. These data sets are used for a single cloning done by the stored procedure. | No | No | Not selected | |
| Data set prefix The prefix for the subsystem cloning stored procedure cloning data sets. These data sets will be used by the subsystem cloning stored procedure to hold the JCL and status of an individual cloning. | No | No | DB2TOOL.CKZ.SP. CLONE | |
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Data set VOLSER The volume serial number (VOLSER) for the subsystem cloning stored procedure data sets. If the data sets will be managed by SMS, leave this field blank. | No | No | blank | |

Table 13. Plan bind, plan name, and other parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|----------------------|------------|
| Data set type The data set type for the subsystem cloning stored procedure data sets. For a PDS, specify PDS. For a PDSE, specify LIBRARY. | No | No | LIBRARY | |
| Create Subsystem Cloning stored procedure product parameters Creates the subsystem cloning stored procedure product parameters member in the parameters library. One job will be generated. This job will create the PPARM member in the prefix.PARMS data set. | No | No | Not selected | |
| Data set prefix The prefix for the subsystem cloning stored procedure parameters data set. | No | No | DB2TOOL.CKZ.SP | |
| Create Subsystem Cloning stored procedure Db2 system parameters Creates the subsystem cloning stored procedure Db2 system parameters member in the parameters library. One job per Db2 entry will be generated. This job adds Db2 system parameters for the selected Db2 entry to the SPARM member of the prefix.PARMS data set. | No | No | Not selected | |
| Data set prefix The prefix for the subsystem cloning stored procedure parameters data set. | No | No | DB2TOOL.CKZ.SP | |
| Allocate the library for the stored procedure REXX EXEC Allocates the EXEC library that holds the EXEC used to invoke the subsystem cloning stored procedure. One job will be generated. | No | No | Not selected | |
| EXEC data set name The data set name of the EXEC library where the stored procedure REXX EXEC is to be placed. | No | No | DB2TOOL.CKZ. EXEC | |

Table 13. Plan bind, plan name, and other parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|----------------------|------------|
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Data set VOLSER The volume serial number (VOLSER) for the stored procedure EXEC data set. If the data set will be managed by SMS, leave this field blank. | No | No | blank | |
| Data set type The data set type for the stored procedure EXEC data set. For a PDS, specify PDS. For a PDSE, specify LIBRARY. | No | No | LIBRARY | |
| Create Subsystem Cloning stored procedure REXX EXEC Creates the REXX EXEC that can be used to run the subsystem cloning stored procedure. One job will be generated. | No | No | Not selected | |
| EXEC data set name The data set name of the EXEC library where the stored procedure REXX EXEC is to be placed. | No | No | DB2TOOL.CKZ. EXEC | |
| Create job to run Subsystem Cloning stored procedure Creates a job to run the subsystem cloning stored procedure using the REXX EXEC. One job per Db2 entry will be generated. | No | No | Not selected | |
| EXEC data set name The data set name of the EXEC library where the stored procedure REXX EXEC is to be placed. | No | No | DB2TOOL.CKZ. EXEC | |

Task: Application Cloning (Table Space Cloning) tasks

Description

This set of tasks binds the Application Cloning (Table Space Cloning) packages and allocates the Application Cloning (Table Space Cloning) runtime repository data sets. All Db2 Cloning Tool packages are bound using package versions.

During the customization process, you will enter these values on panel CCQPPRD.

Some of these tasks are *required* and some are *optional*.

Jobs generated

This task generates the following jobs using the listed templates:

Table 14. Generated jobs for Application Cloning (Table Space Cloning) tasks

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|--|---|
| ssBTSdd | CKZBTS | Db2 Cloning Tool Table Space Cloning package binds | BIND authority on the Db2 system, and read access to SCKZDBRM |
| ssTS1 | CKZTS1 | Allocate Db2 Cloning Tool Table Space Cloning runtime repository data sets | Authority to create the data sets |
| ssTS2 | CKZTS2 | Allocate Db2 Cloning Tool Table Space Cloning data sets | Authority to create the data sets |
| ssRLAdd | CKZRLA | Drop the log apply table | Authority to DROP the log apply table and associated database, table space, and indexes |
| ssDLAdd | CKZDLA | Create the log apply table | Authority to CREATE the log apply table and associated database, table space, and indexes |
| ssALAdd | CKZALA | Upgrade the log apply table for Db2 V11 support | Authority to ALTER the log apply table |

Required authority

Refer to the job list for the required authorizations.

Table 15. Bind and allocation tasks

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|-------------------------|------------|
| Run Application Cloning (Table Space Cloning) package binds Binds the Db2 Cloning Tool Table Space Cloning packages. One job per Db2 entry will be generated. | Yes | No | Selected | |
| Allocate Application Cloning (Table Space Cloning) runtime repository data sets Allocates the Db2 Cloning Tool Table Space Cloning runtime repository data sets. One job will be generated. The prefix.RRJOB data set and prefix.RRDSN data set are created. | No | No | Not selected | |
| Data set prefix The prefix for the Db2 Cloning Tool Table Space Cloning runtime repository data sets. | No | No | DB2TOOL.CKZ. TSREPOS | |

Table 15. Bind and allocation tasks (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|----------------------|------------|
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Data set VOLSER The volume serial number (VOLSER) to be used for the Db2 Cloning Tool Table Space Cloning repository data sets. If data sets will be managed by SMS, leave this field blank. | No | No | blank | |
| Allocate Application Cloning (Table Space Cloning) data sets Allocates the Db2 Cloning Tool Table Space Cloning data sets. One job will be generated. The following data sets are created: prefix.CKZIN, prefix. CKZINTRB, prefix.CKZTIRI, prefix.CKZTIRO, prefix.CKZTRPI, prefix.CKZTRPO, prefix.CMDSSTPT, prefix.CMDSSTPS, prefix.CMDSSTRS, prefix.COPYDSNS, prefix.DDLOUT, prefix.IDCAMS, prefix.LISTDEF, prefix.LOGAPCTL, prefix.MASKDEF, prefix.RDREPL, prefix.RDREPS, prefix.RRJREPL, prefix.RRJREPS, prefix.SYNCDB2, prefix.TDDIN, prefix.TDDOUT, prefix.ULOUT, prefix.ULPUN, prefix.ULREC, prefix.ULUT | No | No | Not selected | |
| Data set prefix The prefix for the Db2 Cloning Tool Table Space Cloning data sets. | No | No | DB2TOOL.CKZ. TSDS | |
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Data set VOLSER The volume serial number (VOLSER) for the table space cloning data sets. If the data sets will be managed by SMS, leave this field blank. | No | No | blank | |

Table 15. Bind and allocation tasks (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| Data set type The data set type for the table space cloning data sets. For a PDS, specify PDS. For a PDSE, specify LIBRARY. | No | No | LIBRARY | |
| Drop Application Cloning (Table Space Cloning) log apply table Drops the log apply table and associated database, table space, and indexes. One job per Db2 entry will be generated. | No | No | Not selected | |
| Create Application Cloning (Table Space Cloning) log apply table Creates the log apply table and associated database, table space, and indexes. One job per Db2 entry will be generated. | Yes | No | Selected | |
| Upgrade Application Cloning (Table Space Cloning) log apply table for Db2 V11 support Upgrades the log apply table for Db2 V11 support. One job per Db2 entry will be generated. | No | No | Not selected | |

Task: ISPF interface tasks

Description

This set of tasks binds the Db2 Cloning Tool interface package, creates the ISPF interface CLISTs, and allocates data sets that are used by the ISPF interface. All Db2 Cloning Tool packages are bound using package versions. During the customization process, you will enter these values on panel CCQPPRD.

These tasks are *required* if you are planning to use the ISPF interface.

Note: PTF PH09578 contains extensive changes to the Db2 Cloning Tool ISPF panels, specifically for table space cloning (now called application cloning). Existing Db2 Cloning Tool customers can bypass installation of the new ISPF interface that is implemented in PTF PH09578 by following the instructions as noted in the step and task descriptions in the table. New customers can run the existing ISPF interface by modifying the CLIST after the jobs have been run; see the instructions in [Create ISPF interface CLISTs](#).

Jobs generated

This task generates the following jobs using the listed templates:

Table 16. Generated jobs for ISPF interface tasks

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|---|-----------------------------------|
| ssPF0 | CKZPF0 | Allocate the ISPF interface CLIST library | Authority to create the data sets |

Table 16. Generated jobs for ISPF interface tasks (continued)

| Jobname | Template | Description | Required authority to run the job |
|----------|----------|---|--|
| ssPF1 | CKZPF1 | Create the CLISTs that are used to invoke the ISPF interface. Note: To continue to use the existing Db2 Cloning Tool ISPF interface, existing customers should not submit this job. New customers can run the existing ISPF interface by modifying the CLIST after it has been created. | Authority to update the CLIST library data set |
| ssPF2 | CKZPF2 | Allocate ISPF repository data sets | Authority to create the data sets |
| ssPF3 | CKZPF3 | Allocate ISPF control data set | Authority to create the data set |
| ssIDBSdd | CKZIDBS | Store Db2 parameters in control data set. | Authority to read and write to the ISPF interface control data set. |
| ssPF4 | CKZPF4 | Migrate table space cloning ISPF profiles to the application cloning profiles that are required for the new ISPF UI implemented in APAR PH09578. Note: Do not submit this job if you want to continue to use the existing Db2 Cloning Tool ISPF interface. | Authority to read and write to table space cloning profiles and to table space cloning profile data sets |

Required authority

Refer to the job list for the required authorizations.

Table 17. ISPF interface parameters

| Step or parameter | Required? | Discovered? | Default value | Your value |
|--|-----------------------------|-------------|---------------|------------|
| ISPF interface Binds the ISPF interface packages, creates the ISPF interface CLISTs and allocates the data sets that are used by the ISPF interface. | If using the ISPF interface | No | Selected | |
| Run ISPF interface package binds Performs a Db2 BIND for the ISPF interface packages. One job per Db2 entry will be generated. | Yes | No | Not selected | |

Table 17. ISPF interface parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|-----------------------|------------|
| Allocate the ISPF interface CLIST library Allocates the CLIST library that holds the CLISTs used to invoke the ISPF interface. One job will be generated. | Yes | No | Not selected | |
| CLIST data set name The data set name of the CLIST library where the ISPF interface CLISTs are to be placed. | Yes | No | DB2TOOL.CKZ. CLIST | |
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Data set VOLSER The volume serial number (VOLSER) for the ISPF interface CLIST data set. If the data set will be managed by SMS, leave this field blank. | Yes | No | blank | |
| Data set type The data set type for the ISPF interface CLIST data set. For a PDS, specify PDS. For a PDSE, specify LIBRARY. | Yes | No | LIBRARY | |
| Create ISPF interface CLISTs Creates the CLISTs that are used to invoke the ISPF interface. One job will be generated. Note: To continue to use the existing Db2 Cloning Tool ISPF interface, existing customers should not select this task. New customers can run the existing ISPF interface by modifying the CLIST after this job has been run; in the CKZCLIST, change CKZWMAIN to CKZ2MAIN. | Yes | No | Not selected | |
| CLIST data set name The data set name of the CLIST library where the ISPF interface CLISTs are to be placed. | Yes | No | DB2TOOL.CKZ. CLIST | |

Table 17. ISPF interface parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|--------------------------|------------|
| ISPF control data set prefix The prefix for the ISPF interface control data set. This data set contains the ISPF interface Db2 information. | Yes | Yes | DB2TOOL.CKZ. ISPFCONT | |
| ISPF repository data sets prefix Specifies the prefix for the ISPF interface repository data sets. These data sets contain the ISPF interface cloning profiles. | Yes | No | DB2TOOL.CKZ. ISPFREPO | |
| Data set VOLSER The volume serial number (VOLSER) to be used for allocating the data set. If the data set will be managed by SMS, leave this field blank. | No | No | blank | |
| Unit name The unit name to use for allocating the data set. The name must be 8 characters or less. | Yes | No | SYSDA | |
| Allocate ISPF repository data sets Performs allocation of the repository data sets to be used by the ISPF interface. One job will be generated. The prefix.PROFILES data set and prefix.PROFILE.RPT data set are created. The repository data sets contain the ISPF interface cloning profiles. | No | No | Not selected | |
| ISPF repository data sets prefix Specifies the prefix for the ISPF interface repository data sets. These data sets contain the ISPF interface cloning profiles. | Yes | No | DB2TOOL.CKZ. ISPFREPO | |
| Data set VOLSER The volume serial number (VOLSER) to be used for allocating the data set. If the data set will be managed by SMS, leave this field blank. | No | No | blank | |

Table 17. ISPF interface parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|--------------------------|------------|
| <p>Allocate ISPF control data set Allocates the control data set to be used by the ISPF interface. One job will be generated. The prefix.CONTROL data set is created. The control data set contains the ISPF interface Db2 subsystem information.</p> | No | No | Not selected | |
| <p>ISPF control data set prefix Specifies the prefix for the ISPF interface control data set. This data set contains the ISPF interface Db2 information.</p> | Yes | No | DB2TOOL.CKZ. ISPFCONT | |
| <p>Data set VOLSER The volume serial number (VOLSER) to be used for allocating the data set. If the data set will be managed by SMS, leave this field blank.</p> | No | No | blank | |
| <p>Store Db2 parameters in ISPF control data set This step saves Db2 subsystem information to the control data set to be used by the ISPF interface. One job per Db2 entry will be generated.</p> | No | No | Not selected | |
| <p>Action if Db2 subsystem already exists This parameter indicates the action to take during the import of Db2 subsystem information from Tools Customizer panels to the ISPF control data set when a Db2 subsystem already exists in the ISPF control data set. REPLACE means that Db2 subsystem information will be overwritten in the ISPF control data set. SKIP means that Db2 subsystem information will not be updated in the ISPF control data set.</p> | No | No | SKIP | |
| <p>ISPF repository data sets prefix This parameter specifies the prefix for the ISPF interface repository data sets. These data sets contain the ISPF interface cloning profiles.</p> | Yes | No | DB2TOOL.CKZ. ISPFREPO | |

Table 17. ISPF interface parameters (continued)

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|--------------------------|------------|
| ISPF control data set prefix This parameter specifies the prefix for the ISPF interface control data set. This data set contains the ISPF interface Db2 subsystem information. | Yes | No | DB2TOOL.CKZ. ISPFCNT | |
| Migrate ISPF repository profiles This step performs migration of table space cloning profiles to the application cloning profiles that are required for the new UI implemented in APAR PH09578. One job will be generated. Note: Do not select this task if you want to continue to use the existing Db2 Cloning Tool ISPF interface. | Yes | No | Not selected | |
| ISPF repository data sets prefix Specifies the prefix for the ISPF interface repository data sets. These data sets contain the ISPF interface cloning profiles. | Yes | No | DB2TOOL.CKZ. ISPFREPO | |

Task: Installation Verification Process

Description

This task creates a Db2 Cloning Tool Table Space Cloning profile that can be used as part of an installation verification process (IVP). During the customization process, you will enter these values on panel CCQPPRD.

This task is not required.

Jobs generated

This task generates the following jobs using the listed templates:

Table 18. Generated jobs for the Installation Verification Process task

| Jobname | Template | Description | Required authority to run the job |
|---------|----------|---|---|
| ssIVPdd | CKZIVP | Creates a table space cloning profile that can be used as part of the IVP | Authority to create or update a Db2 Cloning Tool Table Space Cloning profile. |

Required authority

Refer to the job list for the required authorizations.

Table 19. Installation verification process parameters

| Step or parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Installation Verification Process Creates an Installation Verification Process table space cloning profile. | No | No | Not selected | |
| Create IVP Application Cloning (Table Space Cloning) profile Creates a table space cloning profile for the Installation Verification Process. | Yes | No | Selected | |

Db2 Parameters section

Description

This section contains Db2 parameters. All parameters are required. During the customization process, you will enter these values on panel CCQPDB2.

Table 20. Db2 parameters

| Parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| Db2 subsystem ID description A description for the Db2 subsystem. The value must be 72 characters or less. | No | Yes | blank | |
| Group attach name The group attach name. | No | Yes | blank | |
| General Db2 Information - common | | | | |
| Location name The Db2 location name. The value must be 16 characters or less. | Yes | Yes | LOC1 | |
| Mode The mode in which the Db2 subsystem is running. The following values are valid: <ul style="list-style-type: none"> • CM is compatibility mode on all listed Db2 versions except Db2 10. • CM8 is conversion mode from Db2 V8 on Db2 10. • CM9 is conversion mode from Db2 Version 9.1 on Db2 10. • NFM is new function mode on all listed Db2 versions. | Yes | Yes | NFM | |

Table 20. Db2 parameters (continued)

| Parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------------|------------|
| Level number The version, release, and modification level of the Db2 subsystem. The following values are valid: <ul style="list-style-type: none"> • 910 is valid only for CM or NFM. • 101 is valid only for CM8, CM9 or NFM. • 111 is valid only for CM or NFM. • 121 is valid only for CM or NFM. | Yes | Yes | blank | |
| Db2 Libraries - common | | | | |
| Load library This parameter indicates the data set name of the Db2 load library. | Yes | Yes | DSN.SDSNLOAD | |
| Run library This parameter indicates the data set name of the Db2 run library. | Yes | Yes | DSN.RUNLIB. LOAD | |
| Exit library This parameter indicates the data set name of the Db2 exit library. | Yes | Yes | DSN.SDSNEXIT | |
| Bootstrap data set This parameter indicates the name of the Db2 bootstrap data set. | Yes | Yes | DSN. SDSNBSDS | |
| Db2 Utilities - common | | | | |
| SYSAFF for Db2 utilities This parameter indicates the SYSAFF job parameter to use for running Db2 batch utility jobs. The value must be 8 characters or less. | No | Yes | blank | |
| Plan name for the DSNTIAD utility This parameter indicates the plan name for the DSNTIAD utility. The value must be 8 characters or less. | Yes | No | DSNTIAD | |
| Db2 Tools Objects - common | | | | |
| Storage group name The name of the storage group that will be used for creating Db2 objects for customization. The value must be 128 characters or less. | Yes | No | SYSDEFLT | |

Table 20. Db2 parameters (continued)

| Parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| Schema name The name of the schema that will be used for creating Db2 objects for customization. The value must be 128 characters or less. | Yes | No | SYSTOOLS | |
| Plan or package owner The name of the owner of the plan or the package in the BIND command. | Yes | No | DB2USER | |
| SQL authorization ID The SQL authorization ID that will be used to create Db2 objects. | Yes | No | DB2USER | |
| Db2 Cloning Tool parameters - common | | | | |
| Db2 package name The Db2 package name to use for Db2 Cloning Tool. This package name is used for both Db2 Cloning Tool Subsystem Cloning and Db2 Cloning Tool Table Space Cloning packages. | Yes | No | CKZPACK | |
| Db2 APPLCOMPAT bind option Specify this parameter to generate the APPLCOMPAT option explicitly in the BIND PACKAGE statements. If no value is specified, the APPLCOMPAT option will not be generated explicitly in the BIND PACKAGE statements, and the value from the Db2 ZPARM will be used as the default. This parameter should only be specified when you must keep the value in your Db2 ZPARM, but want to use a different value for APPLCOMPAT. Valid values include V10R1 for Db2 11 CM and V11R1 for Db2 11 NFM. The recommended value for Db2 12 is the highest activated function level. | No | No | blank | |
| Normal DSNZPARM name The normal DSNZPARM name. This is the name of the DSNZPARM to be used when starting the Db2 subsystem in normal mode. | Yes | Yes | blank | |

Table 20. Db2 parameters (continued)

| Parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| Data sharing group name The data sharing group name. This is the name of the data sharing group that this Db2 subsystem is a member of. If not part of a data sharing group, leave this field blank. | No | Yes | blank | |
| Data sharing member name The data sharing member name. This is the member name of this Db2 subsystem in the data sharing group. If not part of a data sharing group, leave this field blank. | No | Yes | blank | |
| Use Subsystem as Source or Target only Indicates whether the Db2 subsystem that is being defined can be used as a source, a target, or either. If you specify SOURCE or TARGET, the subsystem will only appear in the appropriate source or target selection list when the list is presented. If the Db2 subsystem can be used as both a source or a target, enter EITHER. | No | Yes | EITHER | |
| Db2 Cloning Tool Subsystem Cloning parameters | | | | |
| Buffer pool for index spaces Specifies the buffer pool name for additional indexes on the Db2 catalog tables to help improve the performance of the DB2SCHEMA-UPDATE command. | No | No | BPO | |
| Db2 VSAM catalog name The Db2 VSAM catalog name. This is the high level qualifier used for the Db2 directory and Db2 catalog data sets. | No | Yes | blank | |
| Special DSNZPARM name The special DSNZPARM name. This is the name of the DSNZPARM to be used when starting the Db2 subsystem in special mode. | No | Yes | blank | |
| DDF generic LUNAME The DDF generic LUNAME. This parameter corresponds to the GENERIC keyword of the DB2UPDATE command. | No | Yes | blank | |

Table 20. Db2 parameters (continued)

| Parameter | Required? | Discovered? | Default value | Your value |
|--|-----------|-------------|---------------|------------|
| DDF LUNAME The DDF LUNAME. This parameter corresponds to the LUNAME keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF password The DDF password. This parameter corresponds to the PASSWORD keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF TCP/IP port number The DDF TCP/IP port number. This parameter corresponds to the PORT keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF TCP/IP resynchronization port number The DDF TCP/IP resynchronization port number. This parameter corresponds to the RESPORT keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF TCP/IP secure port number The DDF TCP/IP secure port number. This parameter corresponds to the SECPORT keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF TCP/IP IP name The DDF TCP/IP IP name. This parameter corresponds to the IPNAME keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF IPV4 address The DDF TCP/IP IPV4 address. This parameter corresponds to the IPV4 keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF IPV6 address The DDF TCP/IP IPV6 IP address. This parameter corresponds to the IPV6 keyword of the DB2UPDATE command. | No | Yes | blank | |

Table 20. Db2 parameters (continued)

| Parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| DDF group IPV4 address The DDF TCP/IP group IPV4 address. This parameter corresponds to the GRPIPV4 keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF group IPV6 address The DDF TCP/IP group IPV6 address. This parameter corresponds to the GRPIPV6 keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF static alias name The DDF static alias name. This parameter corresponds to the ALIAS keyword of the DB2UPDATE command. | No | Yes | blank | |
| DDF dynamic alias name The DDF dynamic alias name. This parameter corresponds to the DYNAMIC-ALIAS keyword of the DB2UPDATE command. | No | Yes | blank | |
| Db2 Cloning Tool Application Cloning parameters | | | | |
| Database name for log apply table The database name for the table space cloning log apply table. | Yes | No | SYSTOOLS | |
| Buffer pool for log apply table The buffer pool name for the table space cloning log apply table index. | Yes | No | BPO | |
| Buffer pool for log apply table index The buffer pool name for the table space cloning log apply table. | Yes | No | BPO | |
| Data sharing member ID The member ID of this Db2 subsystem in the data sharing group. If not part of a data sharing group, leave this value blank. | No | No | blank | |
| Default VSAM catalog name The default high level qualifier to be used for the target data sets if the target Db2 table spaces and index spaces are not pre-defined. | No | No | blank | |
| Db2 Cloning Tool Subsystem Cloning stored procedure parameters | | | | |

Table 20. Db2 parameters (continued)

| Parameter | Required? | Discovered? | Default value | Your value |
|---|-----------|-------------|---------------|------------|
| Stored procedure name The name for the subsystem cloning stored procedure. This is the unqualified part of the stored procedure name. | Yes | No | CLONE_SS | |
| Stored procedure WLM environment name The WLM environment name for the subsystem cloning stored procedure. To use the Db2 system default WLM environment, leave this field blank. | No | No | blank | |
| Db2 Cloning Tool Installation Verification Process parameters | | | | |
| Use sample tables for Db2 version Specifies the Db2 version for the selection of sample tables that will be used in the Installation Verification Process table space cloning profile. Valid values are CURRENT, 10, 11, and 12. If CURRENT is specified, then the current version of Db2 is used to select the sample tables. If 10 is specified, then the DSN8D10A sample database is used. If 11 is specified, then the DSN8D11A sample database is used. If 12 is specified, then the DSN8D12A sample database is used. | Yes | No | CURRENT | |
| Sample schema name If you want the schema name to be different from the standard Db2 sample schema name of DSN8v10 (where v is a letter representing the version and is dependent on value that is specified in Use sample tables for Db2 version field), enter the schema name. | No | No | None | |
| Db2 Cloning Tool Subsystem Cloning TCP/IP server parameters | | | | |
| TCP/IP port number The default value for the TCP/IP port number on which the TCP/IP server job/started task will be listening. Valid values are 1 - 65535. | No | No | 5099 | |

| <i>Table 20. Db2 parameters (continued)</i> | | | | |
|--|------------------|--------------------|----------------------|-------------------|
| Parameter | Required? | Discovered? | Default value | Your value |
| TCP/IP IP name The default value for the DNS name of the LPAR on which the TCP/IP server job/started task will be running. | No | No | blank | |
| IPV4 address The default dotted decimal IP address, if the IPV4 protocol is used for the connection to the TCP/IP server job/started task. | No | No | blank | |
| IPV6 address The default colon hexadecimal IP address, if IPV6 protocol is used for the connection to TCP/IP server job/started task. | No | No | blank | |

LPAR Parameters section

Description

This section contains LPAR parameters. All parameters are required. During the customization process, you will enter these values on panel CCQPLPR.

| <i>Table 21. LPAR parameters</i> | | | | |
|---|------------------|--------------------|----------------------|-------------------|
| Parameter | Required? | Discovered? | Default value | Your value |
| Job Entry Subsystem (JES) | | | | |
| JES version The version of the job entry system (JES) that is installed on the LPAR. The value must be one character or less. | Yes | Yes | 2 | |

Chapter 3. Starting and preparing Tools Customizer for use

Use the provided REXX EXEC to start Tools Customizer. The first time that you use Tools Customizer, you must modify the settings that Tools Customizer uses to customize Db2 Cloning Tool.

Starting Tools Customizer

Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.

Before you begin

Tools Customizer must be SMP/E installed. You must know the high-level qualifier of where the Tools Customizer libraries reside. The high-level qualifier is considered to be all the segments of the data set name except the lowest-level qualifier, which is SCCQEXEC.



Attention: Ensure that Tools Customizer load libraries are not APF authorized. APF authorizing Tools Customizer libraries results in an abend.

About this task

To run the REXX EXEC, you must either change the placeholder in the EXEC for the high-level qualifier of the Tools Customizer EXEC library or pass the high-level qualifier as a parameter when you run the EXEC. The REXX EXEC is in the CCQTCZ member of the EXEC library.

Procedure

1. Optional: Change the placeholder for the high-level qualifier in the REXX EXEC:
 - a) Find the EXEC library data set for Tools Customizer. The name of the data set is *high_level_qualifier.SCCQEXEC*.
 - b) Edit data set member CCQTCZ and replace the <TCZ HLQ> string with the high-level qualifier of the EXEC library data set.
For example, if the name of the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC, replace <TCZ HLQ> with CCQTCZ.USABSAND.

You have to change the placeholder for the high-level qualifier only once. When you run the REXX EXEC, you do not have to pass the high-level qualifier as a parameter.

2. Run the REXX EXEC (CCQTCZ):
 - a) From the **ISPF Primary Option Menu**, select option 6.
The **ISPF Command Shell** panel is displayed.
 - b) Specify the EX command to run the REXX EXEC.
For example, if the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC and you changed the placeholder for the high-level qualifier in the REXX EXEC, specify:

```
EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'
```

If you did not change the placeholder for the high-level qualifier in the REXX EXEC, specify:

```
EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)' 'CCQTCZ.USABSAND'
```

You can also specify a trace data set name and a user profile when you run the REXX EXEC.

- The default trace data set name is *USERID.CCQ.TRACE*. To specify a different trace data set name, append the trace data set name to the command. For example, to specify a trace data set name of *CCQTCZ.MYTRACE*, enter:

```
EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)' 'CCQTCZ.USABSAND, CCQTCZ.MYTRACE'
```

- To specify a user profile other than your own, append the user profile name to the command. For example, to specify a user profile of *SHRPROF*, enter:

```
CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)' 'CCQTCZ.USABSAND, ,CCQTCZ.SHRPROF'
```

Tools Customizer will use the settings from the specified profile. This profile will be updated when you exit Tools Customizer, but your own profile will remain unchanged.

Results

The **IBM Customizer Tools for z/OS** main menu panel is displayed.

What to do next

If you are running Tools Customizer for the first time, you must modify the Tools Customizer user settings. If you have already set the Tools Customizer user settings, either customize or recustomize Db2 Cloning Tool.

Modifying Tools Customizer user settings

Before you can customize Db2 Cloning Tool with Tools Customizer, you must review the settings that Tools Customizer uses. You might have to change the default values to suit your environment. In most cases, you can change the Tools Customizer values at any time. For example, after you have customized Db2 Cloning Tool and are customizing a different product or solution pack, you might have to change the settings.

Procedure

1. On the **IBM Tools Customizer for z/OS** main panel (CCQPHME), specify option 0, **User settings for Tools Customizer**.

The **Tools Customizer Settings** panel (CCQPSET) is displayed, as shown in the following figure:

```
CCQPSET                Tools Customizer Settings                14:03:51
Command ==>
Enter the settings for customizing a product or press End to save and exit.

Commands: SAVE  OPTIONS

Product Customization Settings

*Customization library qualifier . . DB2TOOL.PRODUCT.CUST
  Volume serial . . . . .
*Use DB2 group attach . . . . . YES (YES/NO)

Tools Customizer Library Settings

*Metadata library . . . . . DB2TOOL.CCQ110.SCCQDENU
*Discover output data set . DB2TOOL.CCQ110.DISCOVER
  Volume serial . . . . .
*Data store data set . . . . . DB2TOOL.CCQ110.DATAST01
  Volume serial . . . . .

User Job Card Settings for Customization Jobs

==> //                JOB
(ACCOUNT), 'NAME',
```

Figure 1. The **Tools Customizer Settings** panel (CCQPSET)

Note: An asterisk next to a field indicates that the field is required.

2. Review the values for the following required fields. Use the default value or specify your own value. You must have appropriate READ and WRITE access to the data sets that are specified.

Customization library qualifier

The high-level qualifier that is used as the prefix for the customization library. The customization library is a data set in which the generated jobs to customize Db2 Cloning Tool are stored. WRITE access to this qualifier is required.

For each product to be customized, the first value that is specified for the qualifier is always used, even if you change it after you have generated the customization jobs. For example, if you customize a product and then specify a new qualifier for recustomization, although the new qualifier is saved and displayed, the original value is used.

To maintain multiple instances of Tools Customizer, specify a unique customization library qualifier for each instance of Tools Customizer.

Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Volume serial

The volume name in which the customization library will reside. If you don't specify a volume name, it will be assigned by the system.

Use DB2 group attach

Determines the value that is used in the CONNECT statements in the generated customization jobs. Specify YES for data sharing environments, which causes the group attach name to be used. Specifying NO, in most cases, causes the SSID to be used in the Db2 CONNECT statement.

Important: This field has no effect when you are customizing a product on a Db2 subsystem that is not a member of a data sharing group. In this case, the Db2 subsystem ID (SSID) is always used in the CONNECT statements in the generated customization jobs.

When you are customizing a product on a Db2 subsystem that is a member of a data sharing group, how the Db2 subsystem is defined and the value of the **Use DB2 group attach** field determines the value that is used in the CONNECT statements in the generated jobs. The following table shows whether the SSID or the group attach name is used:

Table 22. The effect of the value of the Use DB2 group attach field in a data sharing environment

| Db2 subsystem definition | Value of the Use DB2 group attach field | Value that is used in the CONNECT statements |
|--|--|---|
| The Db2 subsystem is defined with an SSID. | Yes | Group attach name |
| | No | SSID ¹ |
| The Db2 subsystem is not defined with an SSID. | Yes or No | Group attach name |

Note 1: If you generate jobs for multiple Db2 subsystems that are defined with an SSID and belong to the same data sharing group, the SSID of the first Db2 subsystem that is selected is used.

For example, assume that on the **Customizer Workplace** panel, you generated jobs for the following Db2 subsystems:

- DB2C, which is a stand-alone Db2 subsystem
- DB2A, which is a Db2 subsystem that is a member of data sharing group DSG1
- A Db2 subsystem that was not defined with an SSID that is a member of data sharing group DSGA

The following figure shows how these Db2 entries might be listed on the **Customizer Workplace** panel:

```

Associated DB2 Entries and Parameter Status
Line commands: G - Generate jobs E - Edit B - Browse C - Copy R - Remove
Cmd SSID GrpAttch Lvl Mode User ID Date Status Message
DB2C -- 121 NFM SYSADM 2017/11/09 Ready to Customize
DB2A DSG1 121 NFM SYSADM 2017/11/09 Ready to Customize
-- DSGA 121 NFM SYSADM 2017/11/09 Ready to Customize
----- End of DB2 entries -----

```

The following table shows which values are used in the CONNECT statements in the generated jobs, based on the value of the **Use DB2 group attach** field.

Table 23. Value that is used in the CONNECT statements in the generated jobs

| SSID | GrpAttch | Value of the Use DB2 group attach field | Value that is used in the CONNECT statements |
|------|----------|---|--|
| DB2C | -- | Yes | SSID |
| | | No | SSID |
| DB2A | DSG1 | Yes | Group attach name |
| | | No | SSID |
| -- | DSGA | Yes | Group attach name |
| | | No | Group attach name |

Tools Customizer metadata library

The name of the data set that contains the metadata that is used to display the Db2 and LPAR parameters. The parameters that are displayed on the **LPAR Parameters** panel and the **DB2 Parameters** panel depend on the parameters that you define and the tasks and steps that you select on the **Product Parameters** panel for the product that you are customizing. For example, the Db2 parameters that are required, based on the selected tasks and steps, are displayed on the **DB2 Parameters** panel, and you can edit them. If they are not required, they are not displayed. Read access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Discover output data set

The name of the data set in which the output from the Db2 Cloning Tool Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product, LPAR, and Db2 parameters from a previously customized product. Write access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Volume serial

The volume name in which the discover output data set will reside. If you don't specify a volume name, it will be assigned by the system.

Data store data set

The name of the data set where Tools Customizer stores information about product, LPAR, and Db2 parameter values. Information about which products are associated with which Db2 entries (Db2 subsystems, Db2 group attach names, and Db2 data sharing members) is also stored in this data set. Data set names that exceed 42 characters must be enclosed in single quotation marks ('). The specified data store data set can be used with only one invocation of Tools Customizer at a time. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Volume serial

The volume name in which the data store data set will reside. If you don't specify a volume name, it will be assigned by the system.

User job card settings for customization jobs

The job card information to be inserted into the generated jobs for customizing a product. The default value is the job statement information from the **ISPF Batch Selection** panel.

The first line of the job card automatically begins with the following information:

```
//          JOB
```

where characters 3 - 10 are reserved by Tools Customizer for the job name and includes a blank space after JOB. This name cannot be edited. Information that you specify on the first line of the job card cannot exceed 57 characters. This character limit includes a continuation character. All other lines of the job card cannot exceed 72 characters.

3. Press End to save and exit.

If the Discover output data set and the data store data set that you specified do not exist, Tools Customizer creates them.

Important: If the ISPF sessions unexpectedly ends before you exit Tools Customizer, the fields on the **Tools Customizer Settings** panel (CCQPSET) will be repopulated with default values, and you will be required to review them or specify new values again.

Results

The values are saved, and the **IBM Tools Customizer for z/OS** main menu panel (CCQPHME) is displayed again.

What to do next

You are ready to customize or recustomize Db2 Cloning Tool or to change parameter settings.

Chapter 4. Customizing Db2 Cloning Tool

Using Tools Customizer to customize Db2 Cloning Tool consists of identifying the product to customize; defining any required Db2 Cloning Tool, LPAR, and Db2 parameters; generating the customization jobs; and submitting the jobs.

Customization roadmaps describe the steps that you must complete to customize Db2 Cloning Tool. Separate roadmaps are provided for the three most common types of customizations.

Use the following table to determine which roadmap corresponds to your environment.

Table 24. Customization roadmaps

| Environment description | Roadmap |
|--|---|
| You do not have a customized version of Db2 Cloning Tool, and you need to customize it for the first time. | “Roadmap: Customizing Db2 Cloning Tool for the first time” on page 77 |
| You have already customized a version of Db2 Cloning Tool, and you want to use the same parameter values to customize a different version. | “Roadmap: Customizing a new version of Db2 Cloning Tool from a previous customization” on page 78 |
| You have a customized version of Db2 Cloning Tool, but you want to change one or more parameter values. | “Roadmap: Recustomizing Db2 Cloning Tool” on page 79 |

Roadmap: Customizing Db2 Cloning Tool for the first time

This roadmap lists and describes the steps that are required to customize Db2 Cloning Tool for the first time.

If you are customizing a previous version of Db2 Cloning Tool, see [“Roadmap: Customizing a new version of Db2 Cloning Tool from a previous customization” on page 78](#).

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- The LPAR ISPF libraries that are required to submit the jobs are known.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize Db2 Cloning Tool for the first time.

Table 25. Steps for customizing Db2 Cloning Tool for the first time

| Step | Description | Instructions |
|------|--|---|
| 1 | Specify the product metadata library for the product that you want to customize. The name of this library is <i>hlq.SCKZDENU</i> . | “Specifying the metadata library for the product to customize” on page 80 |
| 2 | Create new Db2 entries and associate them with Db2 Cloning Tool. | “Creating and associating Db2 entries” on page 83 |
| 3 | Define the required parameters. | “Defining parameters” on page 85 |
| 4 | Generate the customization jobs for the product or for the Db2 entries on which Db2 Cloning Tool is ready to be customized. | “Generating customization jobs” on page 90 |

Table 25. Steps for customizing Db2 Cloning Tool for the first time (continued)

| Step | Description | Instructions |
|------|--|--|
| 5 | Submit the generated customization jobs. | “Submitting customization jobs” on page 90 |

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 26. Administrative tasks

| Description | Instructions |
|--|---|
| Browse the different types of parameters. | “Browsing parameters” on page 92 |
| Copy an existing Db2 entry to the list of Db2 entries on which Db2 Cloning Tool can be customized. | “Copying Db2 entries” on page 92 |
| Remove one or more Db2 entries from the associated list. | “Removing Db2 entries” on page 94 |
| Delete one or more Db2 entries from the master list. | “Deleting Db2 entries” on page 94 |
| Display a list of customization jobs that have been previously generated. | “Displaying customization jobs” on page 95 |
| Maintain the customization jobs in the customization library. | “Maintaining customization jobs” on page 95 |

Roadmap: Customizing a new version of Db2 Cloning Tool from a previous customization

This roadmap lists and describes the steps for customizing a new version of Db2 Cloning Tool based on the existing customization values of a previous version of the same product.

Use this roadmap even if the previous version of Db2 Cloning Tool was not customized by using Tools Customizer.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize a new version of Db2 Cloning Tool from a previous customization.

Table 27. Steps for customizing a new version of Db2 Cloning Tool from a previous customization

| Step | Description | Instructions |
|------|---|---|
| 1 | Specify the product metadata library for the product that you want to customize. The name of this library is <i>hlq.SCKZDENU</i> . | “Specifying the metadata library for the product to customize” on page 80 |
| 2 | Use the Db2 Cloning Tool Discover EXEC to discover information about the version of Db2 Cloning Tool that you previously customized manually. | “Discovering Db2 Cloning Tool information automatically” on page 81 |
| 3 | Define the required parameters. | “Defining parameters” on page 85 |

Table 27. Steps for customizing a new version of Db2 Cloning Tool from a previous customization (continued)

| Step | Description | Instructions |
|------|---|--|
| 4 | Generate the customization jobs for the product or for the Db2 entries on which Db2 Cloning Tool is ready to be customized. | “Generating customization jobs” on page 90 |
| 5 | Submit the generated customization jobs. | “Submitting customization jobs” on page 90 |

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 28. Administrative tasks

| Description | Instructions |
|--|---|
| Browse the different types of parameters. | “Browsing parameters” on page 92 |
| Copy an existing Db2 entry to the list of Db2 entries on which Db2 Cloning Tool can be customized. | “Copying Db2 entries” on page 92 |
| Remove one or more Db2 entries from the associated list. | “Removing Db2 entries” on page 94 |
| Delete one or more Db2 entries from the master list. | “Deleting Db2 entries” on page 94 |
| Display a list of customization jobs that have been previously generated. | “Displaying customization jobs” on page 95 |
| Maintain the customization jobs in the customization library. | “Maintaining customization jobs” on page 95 |

Roadmap: Recustomizing Db2 Cloning Tool

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for Db2 Cloning Tool after you have customized it for the first time.

The new customization jobs will replace the customization jobs that were previously generated and stored in the customization library. Part of the recustomization process includes selecting or deselecting optional tasks or steps, changing the definitions of parameters that have already been defined, or both. Use the method in this roadmap instead of deleting customization jobs from the customization library.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.

Complete the steps in the following table to recustomize Db2 Cloning Tool.

Table 29. Required steps for recustomizing Db2 Cloning Tool

| Step | Description | Instructions |
|------|--|---|
| 1 | Specify the product metadata library for the product that you want to recustomize. The name of this library is <i>hlq.SCKZDENU</i> . | “Specifying the metadata library for the product to customize” on page 80 |

Table 29. Required steps for recustomizing Db2 Cloning Tool (continued)

| Step | Description | Instructions |
|------|---|--|
| 2 | Edit the specific tasks, steps, or parameters that need to be changed. | <ul style="list-style-type: none"> • “Defining Db2 Cloning Tool parameters” on page 85 • “Defining LPAR parameters” on page 87 • “Defining Db2 parameters” on page 88 |
| 3 | Generate the customization jobs for the product or for the Db2 entries on which Db2 Cloning Tool is ready to be customized. | “Generating customization jobs” on page 90 |
| 4 | Submit the new generated customization jobs. | “Submitting customization jobs” on page 90 |

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 30. Administrative tasks

| Description | Instructions |
|--|---|
| Browse the different types of parameters. | “Browsing parameters” on page 92 |
| Copy an existing Db2 entry to the list of Db2 entries on which Db2 Cloning Tool can be customized. | “Copying Db2 entries” on page 92 |
| Remove one or more Db2 entries from the associated list. | “Removing Db2 entries” on page 94 |
| Delete one or more Db2 entries from the master list. | “Deleting Db2 entries” on page 94 |
| Display a list of customization jobs that have been previously generated. | “Displaying customization jobs” on page 95 |
| Maintain the customization jobs in the customization library. | “Maintaining customization jobs” on page 95 |

Specifying the metadata library for the product to customize

You must specify a metadata library for the product that you want to customize.

About this task

The product metadata library contains the information that determines which tasks, steps, and parameters are required to customize Db2 Cloning Tool. This information controls what is displayed on the **Product Parameters** panel, the **LPAR Parameters** panel, and the **DB2 Parameters** panel.

After Db2 Cloning Tool has been SMP/E installed, the default name of the product metadata library is *high_level_qualifier*.SCKZDENU, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.

Procedure

1. Specify option 1 on the **Tools Customizer for z/OS** panel.

The **Specify the Product or Pack Metadata Library** panel is displayed. This panel contains a list of the product metadata libraries that you specified most recently. If you are using Tools Customizer for the first time, this list is empty, as shown in the following figure:

```

CCQPHLQ          Specify the Product or Pack Metadata Library          15:12:22
Command ==>>>          Scroll ==>> CSR

Type the name of the metadata library for the product or the pack in the
Metadata library field, or select the library in the list of previous
libraries and press Enter to populate the field. Press Enter to continue.

The default name of the metadata library after the product or pack has been
SMP/E installed is <hlq>.SxxxDENU, where <hlq> is the high-level qualifier for
the product or the pack, and xxx is the three-character prefix for the product
or the pack.

Product or pack metadata library . CKZ.V320.SCKZDENU

Name              Version  Metadata
Library
=>
=>
=>
=>
=>

```

Figure 2. The **Specify the Metadata Library** panel

2. Use one of the following methods to specify the product metadata library:

- Type the name of a fully qualified partitioned data set (PDS) or an extended partitioned data set (PDSE) in the **Product or pack metadata library** field. Double quotation marks (") cannot be used around the name. Single quotation marks (') can be used but are not required. If you are customizing Db2 Cloning Tool for the first time, you must use this method.
- Place the cursor in any column of the Recent Metadata Libraries list, and press Enter to populate **Product or pack metadata library** field. Press Enter again to select product or pack for customization.

Results

If you are customizing Db2 Cloning Tool for the first time, the **Run Discover EXEC** panel is displayed. Otherwise, the **Customizer Workplace** panel is displayed.

What to do next

- Complete the steps that correspond to your environment:

Customizing Db2 Cloning Tool for the first time

Do not run the Db2 Cloning Tool Discover EXEC. Press End. The **Customizer Workplace** panel is displayed. If your environment requires associated Db2 entries, ensure that they are created and associated. If your environment does not require associated Db2 entries, skip this step, and edit Db2 Cloning Tool parameters.

Customizing Db2 Cloning Tool from a previous or current customization

Press Enter to run the Db2 Cloning Tool Discover EXEC. The **Discover Customized Product Information** panel is displayed. Specify the required information for running the EXEC.

Discovering Db2 Cloning Tool information automatically

You can use the Db2 Cloning Tool Discover EXEC to discover information from a previous or current customization of Db2 Cloning Tool.

About this task

Tip: Using the Db2 Cloning Tool Discover EXEC to discover information from a previous or current customization saves time and reduces errors that can occur when parameters are specified manually.

Db2 Cloning Tool provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on Db2 Cloning Tool.

Parameter values that are discovered and parameter values that are specified manually are saved in the data store. If parameter values for the product that you want to customize exist in the data store, Tools Customizer issues a warning before existing values are replaced.

Procedure

1. On the **Customizer Workplace** panel, issue the DISCOVER command.

If you chose to run the Db2 Cloning Tool Discover EXEC on the pop-up panel after you specified the product to customize, skip this step.

Tip: You can run any Tools Customizer primary command by using either of the following methods:

- Place the cursor on the name of the primary command, and press Enter.
- Type the primary command name in the command line, and press Enter.

The **Discover Customized Product Information** panel is displayed, as shown in the following figure:

```
CCQPDSC          Discover Customized Product Information          08:26:35
Command ==>>>          Scroll ==>> CSR

For the product you are customizing, the Discover EXEC retrieves product
information from an already customized product. Specify the required
information. To save your information and run the Discover EXEC, issue the RUN
command. To save your information and stay on this panel, issue the SAVE
command. To verify the syntax of your information without saving it, press
Enter. To save and exit, press End.

Commands: RUN  SAVE

Discover EXEC for Extracting Information from an Already Customized product
*Discover EXEC library . . . CKZ.V320.SCKZDENU
Discover EXEC name . . . . : CKZDISC
*Discover output data set . . TWUSR.TCZCKZ1.DISCOVER

Information for Discover EXEC
Previous release DB2 Cloning Tool CKZINI library
DB2TOOL.CKZ310.SCKZPARM
DB2 Cloning Tool ISPF control data set
```

Figure 3. The **Discover Customized Product Information** panel

2. Either accept the default values for the following input fields that Tools Customizer generates, or replace the default values with your own values:

Discover EXEC library

The fully qualified data set name that contains the Db2 Cloning Tool Discover EXEC.

Discover EXEC name

The name of the Db2 Cloning Tool Discover EXEC.

Discover output data set

The fully qualified data set where output from the Db2 Cloning Tool Discover EXEC is stored.

3. Either accept or change the default values in the **Information for Discover EXEC** fields.
These fields are generated by Db2 Cloning Tool. They show the information that is required to run the Db2 Cloning Tool Discover EXEC.
4. Issue the RUN command to run the Db2 Cloning Tool Discover EXEC.
Alternatively, save your information without running the Db2 Cloning Tool Discover EXEC by issuing the SAVE command.
If you issue the RUN command to run the Db2 Cloning Tool Discover EXEC, the parameter information is discovered for Db2 Cloning Tool, and the **Customizer Workplace** panel is displayed.

Results

The discovered parameter values for Db2 Cloning Tool replace any existing values.

What to do next

The next step depends on your environment:

- If Db2 entries were not discovered, or if you need to customize Db2 Cloning Tool on new Db2 entries, create and associate the entries.
- If Db2 entries were discovered and you want to customize Db2 Cloning Tool on only these entries, define the parameters.

Related tasks

Creating and associating Db2 entries

You can create new Db2 entries and associate them with Db2 Cloning Tool.

Defining parameters

To customize Db2 Cloning Tool, you must define Db2 Cloning Tool parameters, LPAR parameters, and Db2 parameters, if your customization requires Db2 entries.

Creating and associating Db2 entries

You can create new Db2 entries and associate them with Db2 Cloning Tool.

About this task

The list of associated Db2 entries is on the **Customizer Workplace** panel.

Procedure

1. Issue the ASSOCIATE command on the **Customizer Workplace** panel.

The **Associate DB2 Entry for Product** panel is displayed, as shown in the following figure:

```
CCQPDAD          Associate DB2 Entry for Product          DB2 entry deleted
Command ==>>>                                     Scroll ==>> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands:  CREATE - Create new DB2 entries

DB2 Entries
Line commands: A - Associate  C - Copy  D - Delete
Cmd SSID GpAttch
----- End of DB2 entries -----
```

Figure 4. The **Associate DB2 Entry for Product** panel

2. Create Db2 entries.

If you need to associate Db2 entries that are already in the master list, skip this step and go to step 3.

- a) Issue the CREATE command to create one Db2 entry, or issue CREATE *nn* to create multiple Db2 entries, where *nn* is the number of new entries to be created.

The **Create DB2 Entries** panel is displayed, as shown in the following figure:

```

CCQPCDB                                Create DB2 Entries                                Row 1 of 1
Command ==>                               Scroll ==> PAGE

Specify the SSID, the group attach name, or both in the appropriate columns
for each new DB2 entry and press Enter. To create additional entries, issue
the Inn line command, where nn is the number of entries to be inserted, and
press Enter. To cancel, press End.

New DB2 Entries
Line commands: I - Insert into list   R - Remove from list
Cmd  SSID  GrpAttch  Message
-----
----- End of DB2 entries -----

```

Figure 5. The **Create DB2 Entries** panel

- b) In the appropriate columns, specify a Db2 subsystem ID, Db2 group attach name, or Db2 data sharing member name for the Db2 entry that you want to create, and press Enter.

Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

Tips:

- To insert multiple Db2 entries, specify the *Inn* line command, where *nn* is the number of Db2 entries to be inserted.
- You will define specific parameters for these new Db2 entries on the **DB2 Parameters** panel. This panel is displayed after you select these new Db2 entries and issue the line command to generate the jobs, after you issue the primary command to generate the jobs for all associated Db2 entries, or when you manually edit the Db2 parameters.

The **Associate DB2 Entry for Product** panel is displayed, and the new Db2 entry is displayed in the master list, as shown in the following figure:

```

CCQPDAD                                Associate DB2 Entry for Product                                Row 1 to 3 of 3
Command ==>                               Scroll ==> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands:  CREATE - Create new DB2 entries

DB2 Entries
Line commands: A - Associate  C - Copy  D - Delete
Cmd  SSID  GrpAttch
*      *
DBAA  --
DBAB  --
DBAC  --
-----
----- End of DB2 entries -----

```

Figure 6. **Associate DB2 Entry for Product** panel

- c) Repeat steps b and c for each Db2 entry that you want to create.
- d) When you have created all the Db2 entries, associate them with Db2 Cloning Tool, or press End to display the **Customizer Workplace** panel.
3. Associate Db2 entries.
- a) Specify A against one or more Db2 entries in the master list, and press Enter to associate them with Db2 Cloning Tool.

Results

The **Customizer Workplace** panel is displayed with the associated Db2 entries displayed in the associated list.

What to do next

Define the parameters.

Related concepts

[Tools Customizer terminology](#)

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Defining parameters

To customize Db2 Cloning Tool, you must define Db2 Cloning Tool parameters, LPAR parameters, and Db2 parameters, if your customization requires Db2 entries.

About this task

You must define the Db2 Cloning Tool parameters first for the following reasons:

- If you ran the Db2 Cloning Tool Discover EXEC, you must review the values that were discovered.
- If you select optional tasks and steps on the **Product Parameters** panel that affect the Db2 entry on which you will customize Db2 Cloning Tool, additional parameters might be displayed on the **DB2 Parameters** panel.
- If other steps must be completed in a specific sequence, customization notes on the **Product Parameters** panel will display the correct sequence.

Defining Db2 Cloning Tool parameters

Db2 Cloning Tool parameters are specific to Db2 Cloning Tool.

About this task

If you ran the Db2 Cloning Tool Discover EXEC, you must review the parameters that were discovered.

Procedure

1. Specify E next to the **Product parameters** field on the **Customizer Workplace** panel, and press Enter. The **Product Parameters** panel is displayed, as shown in the following figure. If other steps must be completed in a specific sequence before you define the Db2 Cloning Tool parameters, a note labeled **Important** will display the correct sequence on this panel.

```
CCQPPRD          Product Parameters: DB2 Cloning Tool          08:33:17
Command ===>                               Scroll ===> CSR

Complete the following tasks to customize the products. The required tasks,
required steps within a required or selected task, and required parameters
are preceded by an asterisk (*). Ensure that values are specified for the
required parameters. Press End to save and exit.

Commands: SAVE  VERIFYOFF
Line Commands: / - Select

Product customization library : TWUSR.TCZCKZ1.$RS01$.CKZ320          More:      +

Usage Notes:
- If the Subsystem Cloning stored procedure will be used, the DB2
  administrative task scheduler must be configured and available on the DB2
  systems where the stored procedure will be run.

Common parameters
  HLQ of current release . DB2TOOL.CKZ320

* Run INIMERGE
```

Figure 7. The **Product Parameters** panel

You can use the following primary commands on this panel:

SAVE

Saves the specified product or component parameter values.

VERIFY / VERIFYOFF

Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of product or component parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The product or component parameter status of **Verify Values** on the **Customize Workplace** panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the **DB2 Parameters** panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the Product or Component Parameters Values panel.

2. Select any required tasks and steps, and specify values for any parameters. After you select a task or step with a slash (/), put the cursor in the selected field and press Enter.

If tasks, steps, and parameters are required, they are preselected with a slash (/). Otherwise, they are not preselected.

All of the required parameters have default values, which you can either accept or change.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
 - For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
 - The following elements apply to specific fields:
 - **Add** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add**, and press Enter. Use the displayed panel to add or delete additional values.
 - **List** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List**, and press F1 or the key that is mapped to Help.
 - **More** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More**, and press Enter.
3. Optional: Select other tasks and steps with a slash (/) and press Enter to activate the input fields. Either accept or change the default values that are displayed.
 4. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the **Product Parameters** panel.

Results

The **Customizer Workplace** panel is displayed, and the status of the product parameters is Ready to Customize.

What to do next

If the status of other parameters on the **Customizer Workplace** panel is Incomplete, Verify Values, or Discovered, edit these parameters.

Related tasks

[Defining LPAR parameters](#)

LPAR parameters are parameters on the local LPAR that are required to customize Db2 Cloning Tool.

Defining Db2 parameters

Db2 parameters are parameters for a Db2 entry.

Defining LPAR parameters

LPAR parameters are parameters on the local LPAR that are required to customize Db2 Cloning Tool.

Procedure

1. Specify E next to the **LPAR parameters** field, and press Enter.

The **LPAR Parameters** panel is displayed, as shown in the following figure:

Figure 8. The LPAR Parameters panel

You can use the following primary commands on this panel:

SAVE

Saves the specified product or component parameter values.

VERIFY / VERIFYOFF

Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of LPAR parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The LPAR parameter status of *Verify Values* on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the LPAR Parameters panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the LPAR Parameters Values panel.

2. Specify values for all required parameters that are displayed.

Many parameters have default values, which you can either accept or change.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
 - **Add** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add**, and press Enter. Use the displayed panel to add or delete additional values.
 - **List** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List**, and press F1 or the key that is mapped to Help.
 - **More** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More**, and press Enter.

The following LPAR parameters can contain 1 - 64 values:

- LPAR macro library
- Message library
- Panel library
- Skeleton library

- ISPF table input library
 - ISPF user profile library
 - File tailoring output library
 - Link list library
 - Command procedures library
 - Macro library
 - Link-edit library
 - Load library
 - Started task library name
3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results

The **Customizer Workplace** panel is displayed, and the status of the LPAR parameters is Ready to Customize.

What to do next

If the status of other parameters on the **Customizer Workplace** panel is Incomplete, Verify Values, or Discovered, edit these parameters.

Related tasks

[Defining Db2 Cloning Tool parameters](#)

Db2 Cloning Tool parameters are specific to Db2 Cloning Tool.

[Defining Db2 parameters](#)

Db2 parameters are parameters for a Db2 entry.

Defining Db2 parameters

Db2 parameters are parameters for a Db2 entry.

About this task

If you did not run the Db2 Cloning Tool Discover EXEC, you must create and associate one or more Db2 entries before you can define the Db2 parameters. For more information, see [“Creating and associating Db2 entries”](#) on page 83.

Procedure

1. Specify E next to one or more Db2 entries in the associated list, which is in the Associated Db2 Entries and Parameter Status section on the **Customizer Workplace** panel, and press Enter.
The **DB2 Parameters** panel is displayed, as shown in the following figure:

```

CCQPDB2          DB2 Parameters: DB2 Cloning Tool          11:12:56
Command ==>>>          Scroll ==>> CSR

Ensure that values are specified for the required DB2 parameters. Press End
to save and exit.

Commands: SAVE  VERIFYOFF          More:      +

DB2 subsystem ID . . . . . : DB02
DB2 subsystem ID description

Group attach name . . . . .

General DB2 Information - common
*Location name . . . . . LOC1
*Mode . . . . . NFM (CM, CM8, CM9, NFM)
*Level number . . . . . 101 (910, 101, 111)

DB2 Libraries - common
*Load library . . . . . DSN.SDSNLOAD          Add
*Run library . . . . . DSN.RUNLIB.LOAD       Add
*Exit library . . . . . DSN.SDSNEXIT        Add
*Bootstrap data set . . . DSN.SDSNBSDS      Add

```

Figure 9. The **DB2 Parameters** panel

You can use the following primary commands on this panel:

SAVE

Saves the specified product or component parameter values.

VERIFY / VERIFYOFF

Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of Db2 parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The Db2 parameter status of *Verify Values* on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the **DB2 Parameters panel**, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the **DB2 Parameters Values** panel.

2. Specify values for all parameters that are displayed.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
 - **Add** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add**, and press Enter. Use the displayed panel to add or delete additional values.
 - **List** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List**, and press F1 or the key that is mapped to Help.
 - **More** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More**, and press Enter.

Many parameters have default values, which you can either accept or change.

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results

The status of the Db2 entries that you selected on the **Customizer Workplace** panel is Ready to Customize.

What to do next

If the status of other parameters on the **Customizer Workplace** panel is Incomplete, Verify Values, or Discovered, edit these parameters.

Related tasks

[Defining Db2 Cloning Tool parameters](#)

Db2 Cloning Tool parameters are specific to Db2 Cloning Tool.

[Defining LPAR parameters](#)

LPAR parameters are parameters on the local LPAR that are required to customize Db2 Cloning Tool.

Generating customization jobs

To generate customization jobs for Db2 Cloning Tool and any associated Db2 entries, issue the GENERATEALL command, or select one or more Db2 entries on which to customize Db2 Cloning Tool.

Procedure

Generate the customization jobs by using one of the following methods.

- If you want to generate customization jobs at the product level and for any associated Db2 entries, issue the GENERATEALL command, and press Enter.
- If you want to generate customization jobs for specific Db2 entries, select the Db2 entries by specifying the G line command against them, and press Enter. The available Db2 entries are in the associated list in the Associated Db2 Entries and Parameter Status section.

Important: Regenerating customization jobs will replace any existing jobs, including jobs that you might have manually modified after they were generated.

Results

If the status is Incomplete or Discovered for Db2 Cloning Tool parameters, LPAR parameters, or Db2 parameters, Tools Customizer automatically starts an editing session for the types of parameters that are required. The session continues until the panel for each type of required parameter has been displayed.

What to do next

If an automatic editing session is started, accept the displayed parameter values or define values for the required types of parameters, select optional parameters, tasks, or steps for your environment, and save the parameter values. Otherwise, the customization jobs are generated, and you can submit them.

Tip: If the customization jobs are generated, but you are not ready to submit them, you can see them later by issuing the JOBLIST command on the **Customizer Workplace** panel. The JOBLIST command displays the **Finish Product Customization** panel, which you can use to submit the jobs.

Submitting customization jobs

Submit the customization jobs to customize Db2 Cloning Tool.

Before you begin

Ensure that the correct jobs are generated.

About this task

The following figure shows part of the **Finish Product Customization** panel. The table on this panel shows the customization jobs that are generated by Tools Customizer. They are grouped by job sequence number.

```
CCQPCST          Finish Product Customization          Row 1 to 10 of 28
Command ==>                                           Scroll ==> CSR

For a first-time customization, submit the jobs in the members in the order
in which they apply to the DB2 entries. Otherwise, submit only the necessary
jobs that were generated after changes were made. To submit jobs, browse
the members and issue the TSO SUBMIT command.

Line Commands: E - Edit  B - Browse

Product customization library:  TWUSR.TCZCKZ1.$RS01$.CKZ320

Cmd Member  SSID Grpattch Template Date      Description
-----
A0IMR  --  --  CKZIMR  2015/06/17  Run INIMERGE to merge a previou
A1IUP  --  --  CKZIUP  2015/06/17  Run update of CKZINI member
A2BPTAC DB02 --  CKZBPT  2015/06/17  DB2 Cloning Tool plan bind
A3BSSAC DB02 --  CKZBSS  2015/06/17  Subsystem Cloning package bind
A4RSPAC DB02 --  CKZRSP  2015/06/17  Drop Subsystem Cloning stored p
A5DSPAC DB02 --  CKZDSP  2015/06/17  Define Subsystem Cloning stored
A6BSPAC DB02 --  CKZBSP  2015/06/17  Bind the Subsystem Cloning stor
A7RSTAC DB02 --  CKZRST  2015/06/17  Drop Subsystem Cloning global t
A8DSTAC DB02 --  CKZDST  2015/06/17  Define Subsystem Cloning global
A9SP1  --  --  CKZSP1  2015/06/17  Allocate the Subsystem Cloning
```

Figure 10. The **Finish Product Customization** panel

The member-naming conventions depend on whether the customization jobs are for Db2 entries, and LPAR, or the product.

The New filed indicates if the job member is newly created/updated. It is either YES or NO. YES indicates the job member is newly created or updated, and it needs to be submitted for customization. NO indicates the job member is not newly created/updated, it does not need to be submitted for customization.

Customization jobs for Db2 entries

The members use the following naming convention:

```
<job_sequence_number><job_ID><DB2_entry_ID>
```

where

job_sequence_number

Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

job_ID

Characters 4 - 7 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. Db2 Cloning Tool assigns the template name.

DB2_entry_ID

Two alphanumeric characters, AA - 99, that Tools Customizer assigns to a Db2 entry.

For example, the XYZBNDDB2_entry_ID_1 and XYZBNDDB2_entry_ID_2 jobs are generated from the XYZBNDGR template, and the XYZ4DB2_entry_ID_1 and XYZ4DB2_entry_ID_2 jobs are generated from the XYZ4 template. If the jobs are generated on two Db2 entries, the following member names are listed sequentially: A0BNDGAA, A0BNDGAB, A14AA, A14AB.

Customization jobs for an LPAR or the product

The members use the following naming convention:

```
<job_sequence_number><job_ID>
```

where

job_sequence_number

Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

job_ID

Characters 4 - 8 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. For example, for the XYZMAKE template, the job ID is MAKE. For the XYZM template, the job ID is M. Db2 Cloning Tool assigns the template name, and it is displayed in the Template column.

For example, the XYZBNDGR job is generated from the XYZBNDGR template, and the XYZ4 job is generated from the XYZ4 template. The following member names are listed sequentially: A0BNDGR, A14.

Procedure

1. Submit the generated customization jobs by following the process that you use in your environment or by using the following method:
 - a) Specify B or E against a customization job or the product customization library, and press Enter. An ISPF browsing or editing session is started.
 - b) Browse the customization job or each member in the library to ensure that the information is correct.
 - c) Run the TSO SUBMIT command.
2. Press End.

Results

Db2 Cloning Tool is customized, and the **Customizer Workplace** panel is displayed. The status is Customized for the Db2 entries on which Db2 Cloning Tool was customized.

What to do next

You can generate more customization jobs for other Db2 entries, view a list of customization jobs that you previously generated, or recustomize Db2 Cloning Tool.

Browsing parameters

You can browse the product or component parameters, the LPAR parameters, and the Db2 parameters in read-only mode.

Procedure

1. On the **Customizer Workplace** panel, specify B next to the **Product parameters** field, the **LPAR parameters** field, or the Db2 entry that you want to browse, and press Enter. The panel that corresponds to your specification is displayed.
2. Press End to exit.

Copying Db2 entries

You can copy associated and not associated Db2 entries to other Db2 entries or to new Db2 entries.

About this task

Go to the step that applies to your environment:

- To copy an associated Db2 entry to another associated Db2 entry or to an entry that is not associated, go to step 1.
- To copy an associated Db2 entry to a new entry, go to step 2.
- To copy a Db2 entry that is not associated to a new entry, go to step 3.

Procedure

1. To copy an associated Db2 entry to another associated Db2 entry or to an entry that is not associated, complete the following steps:

a) Specify C against a Db2 entry in the associated list of Db2 entries on the **Customizer Workplace** panel, and press Enter.

The **Copy Associated DB2 Entry** panel is displayed.

b) Select one or more Db2 entries to which information will be copied by specifying the / line command, and press Enter.

The Associated column indicates whether the Db2 entry is associated.

Tip: To copy information into all of the Db2 Entries in the list, issue the SELECTALL primary command, and press Enter.

The **Copy DB2 Parameter Values** panel is displayed.

c) Specify an option for copying common and product-specific Db2 parameter values.

Common Db2 parameter values apply to all Db2 entries for all products that you have customized by using Tools Customizer. Product-specific Db2 parameter values apply only to the product that you are currently customizing.

- To copy the common Db2 parameter values and the product-specific Db2 parameter values, specify option 1, and press Enter.

- To copy only the product-specified Db2 parameter values, specify option 2, and press Enter.

In some cases, the Db2 parameter values might contain the Db2 subsystem ID as an isolated qualifier in data set names. For example, in the DB01.DB01TEST.DB01.SANLLOAD, data set name, the DB01 subsystem ID is isolated in the first and third qualifiers but is not isolated in the second qualifier. When the Db2 subsystem ID is an isolated qualifier in data set names, the **Change DB2 Subsystem ID in DB2 Parameter Values** panel is displayed. Otherwise, the **Customizer Workplace** panel is displayed.

d) If the **Change DB2 Subsystem ID in DB2 Parameter Values** panel is displayed, specify an option for changing the subsystem IDs. Otherwise, skip this step.

- To change the subsystem ID in isolated qualifiers in data set names, specify option 1, and press Enter.

- To use the same subsystem ID in all values, specify option 2, and press Enter.

The **Customizer Workplace** panel is displayed with the copied associated entry in the list.

2. To copy an associated Db2 entry to a new entry, complete the following steps:

a) Specify C against a Db2 entry in the associated list of Db2 entries on the **Customizer Workplace** panel, and press Enter.

The **Copy Associated DB2 Entry** panel is displayed.

b) Issue the CREATE command.

The **Create DB2 Entries** panel is displayed.

c) Specify the SSID, the group attach name, or both in the appropriate columns for each new Db2 entry, and press Enter.

Tip: To add rows for additional entries, specify the *Inn* line command, where *nn* is the number of entries to be created, and press Enter.

The **Copy Associated DB2 Entry** panel is displayed with the new entries in the list. The new entries are preselected.

d) Press Enter to complete the copy process.

The **Customizer Workplace** panel is displayed with the copied entries in the list.

3. To copy a Db2 entry that is not associated to a new entry, complete the following steps:

a) Issue the ASSOCIATE command on the **Customizer Workplace** panel.

The **Associate DB2 Entry for Product** panel is displayed.

b) Select one or more Db2 entries by specifying the / line command, and press Enter.

The **Copy a DB2 Entry** panel is displayed.

c) Specify the SSID, the group attach name, or both in the appropriate columns for the new Db2 entry, and press Enter.

The **Associate DB2 Entry for product** panel is displayed with the copied entry in the list.

d) If you want to associate the copied entry, specify A against it, and press Enter.

The **Customizer Workplace** panel is displayed with the copied entries in the list.

What to do next

Edit any of the parameters or generate the jobs.

Related concepts

[Tools Customizer terminology](#)

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Removing Db2 entries

You can remove Db2 entries from the associated list.

About this task

When you remove Db2 entries from the associated list, any customization jobs for the entries are removed from the list of jobs on the **Finish Product Customization** panel, and they are deleted.

Procedure

On the **Customizer Workplace** panel, specify R next to one or more Db2 entries that you want to remove, and press Enter.

The selected Db2 entries are removed from the associated list and added to the master list on the **Associate DB2 Entry for Product** panel, and the customization jobs are deleted.

Related concepts

[Tools Customizer terminology](#)

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Deleting Db2 entries

You can delete Db2 entries from the master list.

About this task

When you delete Db2 entries from the master list, any associations and all customization jobs for products that are customized on the entries will be deleted.

Procedure

1. On the **Customizer Workplace** panel, issue the ASSOCIATE command.

The **Associate DB2 Entry for Product** panel is displayed.

2. Specify D next to one or more Db2 entries that you want to delete, and press Enter.

If the entry is associated with any products, the **Delete Associated DB2 Entry** panel for the first Db2 entry that you selected is displayed. Otherwise, the **Delete DB2 Entry** panel is displayed.

3. To delete the Db2 entries, press Enter.

If the Db2 entries are associated with any products in the table on the **Delete Associated DB2 Entry** panel, any associations and all customization jobs for the products that are customized on it are deleted. Otherwise, only the Db2 entries are deleted. If you selected multiple Db2 entries to delete, the next Db2 entry that you selected is displayed on either the **Delete Associated DB2 Entry** panel or the **Delete DB2 Entry** panel. Otherwise, the **Associate DB2 Entry for Product** panel is displayed.

What to do next

If you selected multiple Db2 entries to delete, repeat step 3 until all selected entries are deleted. Then, continue the customization process.

Displaying customization jobs

You can view a list of the members that contain the customization jobs before or after you submit the jobs.

About this task

The customization jobs that you generate for one Db2 entry are also displayed when you customize Db2 Cloning Tool for another Db2 entry later.

Procedure

On the **Customizer Workplace** panel, issue the JOBLIST command.

The **Finish Product Customization** panel is displayed. This panel shows the list of jobs that you have previously generated. They are grouped by job sequence number. Use this panel to browse or edit the generated jobs before you submit them.

Maintaining customization jobs

Instead of deleting customization jobs outside of Tools Customizer, you can maintain the correct jobs for Db2 Cloning Tool by completing the steps for recustomization.

About this task

You cannot delete or rename customization jobs from the customization library by starting an ISPF browse or edit session from the **Finish Product Customization** panel. If you try to delete customization jobs by using this method, the CCQC034S message is issued. If you try to rename customization jobs, the CCQC035S message is issued.

If you delete or rename customization jobs from the customization library by using ISPF outside of Tools Customizer, Tools Customizer will not recognize that the jobs were deleted, and the **Finish Product Customization** panel will still display them. If you browse or edit jobs that were deleted from the library outside of Tools Customizer, the CCQC027S message is issued.

Procedure

To maintain the correct customization jobs in the customization library, complete the steps for recustomization.

Using Tools Customizer in a multiple-LPAR environment

Currently, Tools Customizer supports only the local LPAR; however, you can propagate customizations to additional LPARs by using either of two different methods.

About this task

In a multiple-LPAR environment, Tools Customizer identifies the LPAR to which you are logged on. Tools Customizer uses this LPAR name for several different parameter settings, one of which is the data store. When you use the data store during the customization of Db2 Cloning Tool that is on a different LPAR, Tools Customizer issues message CCQD586S, which indicates that the product has already been customized based on values from the data store on the first LPAR. This message is issued to prevent the data store from becoming corrupted.

This behavior occurs in the following conditions:

- Tools Customizer is installed on a DASD device that is shared by multiple LPARs.
- After a product is customized by using Tools Customizer, the data store is copied to another LPAR.

Procedure

To customize products running against a Db2 subsystem on an LPAR where Tools Customizer is not installed, consider using one of the following methods:

Install one instance of Tools Customizer on one LPAR

If you intend to reuse the customization values for all the instances of your products on all LPARs, use this method.

- a. Associate all the Db2 entries in this one instance of Tools Customizer. The LPARs on which the Db2 subsystems reside do not matter.
- b. Generate the customization jobs for each Db2 entry.
- c. Copy the generated customization jobs to the LPAR to run against the specific Db2 entries. Some LPAR-specific edits might be required. You can make these edits in the customized jobs that you copied. Note that this situation is one of the few situations where you might need to make manual changes to the jobs that are customized by Tools Customizer.

Install one instance of Tools Customizer on each LPAR

If you do not want to reuse previous customization values and you want to start new customizations, use this method.

Important: This method will likely not be the preferred approach for most organizations because most organizations tend to use similar or identical customization values for each product instance on all LPARs.

Chapter 5. After customizing Db2 Cloning Tool using Tools Customizer

These tasks should be completed after Db2 Cloning Tool is customized, or if you migrate to a new version of Db2 after customizing Db2 Cloning Tool.

Reviewing the CKZINI PARMLIB member variables

The keywords in the CKZINI PARMLIB member provide flexibility and control of product execution, and security of system parameters. Make sure to review and update the generated CKZINI member to ensure all keywords are appropriate for your installation and operation requirements.

This member is a read-only library during execution of the product. You can alter the keyword values by either:

- Using Tools Customizer to update and re-generate the CKZINI member. See the topic [“Task: Create and update CKZINI”](#) on page 33 for more information.
- Directly updating the CKZINI member. [“CKZINI customization values”](#) on page 1144 contains descriptions and coding rules for the keywords in the CKZINI member. Use ISPF Edit to update the CKZINI configuration values.

Some keywords may be coded with default values. You should review the following keywords in particular, but make sure to review all of the CKZINI to ensure the values are correct for your installation:

- Verify the Db2 plan name(s) in the DB2_OPTIONS section. All Db2 subsystems that Db2 Cloning Tool will access must have a plan defined. This plan name is provided to Db2 Cloning Tool using the DB2_PLAN value.
- Verify the CA-MIM/MII token values in RESOURCE_SERIALIZATION section. Installations running CA-MIM/MII with multiple systems and shared DASD need to set the following CKZINI parameter found in the :RESOURCE_SERIALIZATION section to YES to ensure that when CA-MIM/MII GDIF is inactive, the Db2 Cloning Tool data sets are protected from data corruption.

```
:RESOURCE_SERIALIZATION  
MIM_GDIF = YES
```

- Verify the TCP/IP name in the TCPIP-OPTIONS section. If you are planning to use the TCP/IP option of Db2 Cloning Tool Table Space Cloning and your TCP/IP started task name is not TCPIP, update the TCPIP_STC_NAME to match your installation's started task name.

Managing CKZINI when running Db2 Cloning Tool on several systems

For those customers with Db2 Cloning Tool running on several systems, the CKZINI PARMLIB member can be managed either by:

- Using separate copies of the CKZINI, editing them independently and running INIMERGE on each individually.
- Using the same text (multi-image INI) for all systems; this requires replicating sections with sysplex and/or system names qualifications for those sections that need token/values unique to an image. Refer to [“CKZINI customization values”](#) on page 1144 for more information related to qualified section names. Use the CKZIVIEW member of the SCKZJCL data set to view the contents of the multi-image INI that will be used by any image.

Verifying successful customization

When customization is completed, you can verify that Db2 Cloning Tool and Db2 Cloning Tool Table Space Cloning have been successfully installed.

Once the customization tasks have been completed, the FINDUCATS command can be run against one volume to verify that Db2 Cloning Tool has been successfully customized.

To verify that Db2 Cloning Tool Table Space Cloning has been successfully installed, you can use the Installation Verification Process that is described in the topic [“Using the Installation Verification Process”](#) on page 98.

If you have configured the ISPF interface, verify successful customization of the interface by running the CKZ CLIST.

Using the Installation Verification Process

You can use the Db2 Cloning Tool installation verification process (IVP) to verify that basic capabilities of Db2 Cloning Tool Table Space Cloning are installed and working. The IVP creates a table space cloning profile for the ISPF interface that clones objects from the Db2 sample tables that are provided with every version of Db2 for z/OS.

Before you begin

The Db2 sample tables must be installed on the source Db2 subsystem.

About this task

The IVP creates a standard table space cloning profile that clones the Db2 sample tables that are provided with your version of Db2, and imports the profile into the ISPF repository. You then open the sample cloning profile in the ISPF interface, build the source and target jobs from the profile, and run the jobs. Successful completion of these jobs verifies the table space cloning installation.

Several basic functions are set in the sample profile to show and test Db2 Cloning Tool capabilities, including the following:

- The LISTDEF command subsets the objects to be cloned from the sample database. Exclude LISTDEF rules are also used.
- Object translate (database, table, and creator) is specified.
- DDL attribute changes (STOGROUP) are included.
- RI is included (INCLUDE-ALL-RI(Y)).
- The copy uses FUZZY-COPY(Y).
- PROCESS-DDL is enabled with PROCESS-TYPE(Y) and EXPLODE-OBJECTS(Y).
- LOG-APPLY is enabled.
- RTS-COPY is enabled.

Procedure

1. Ensure that the correct Db2 sample tables are installed on the source Db2 subsystem.
2. Run Tools Customizer and do the following:
 - a) On the **Db2 Product Parameters** panel, select the **Installation Verification Process** step and specify the values for the ISPF repository tables and control file.
Refer to [“Worksheets: Gathering parameter values for Tools Customizer”](#) on page 31 for more information.
 - b) On the **Db2 Parameters** panel for the subsystem on which you will run the IVP, do the following:
 - i) Specify the Db2 version of the sample table in the **Use sample tables for Db2 version** field.

- ii) If your schema name is different from the standard Db2 sample schema name, specify your schema name in the **Sample schema name field**. This avoids potential problems during object translation in the source job.
 - iii) Ensure that the subsystem ZPARM has been specified in the required **Normal DSNZPARM name field**.
 - iv) Verify that all required parameters (marked with an asterisk) on the **Db2 Parameters** panel have valid values, especially the parameters for the **Db2 libraries – common group** section.
- c) Generate the Tools Customizer jobs.
- d) Submit the generated CKZIVP job. The job creates a table space cloning profile with the profile name CKZIVP_ssid, where ssid is the connection ID of the source Db2 subsystem.
3. Start the Db2 Cloning Tool ISPF interface.
For more information, refer to [Chapter 22, “Using the ISPF interface,”](#) on page 311.
 4. Locate the CKZIVP_ssid profile under the **Db2 tablespace clone** option (option 1.2 from the main menu).
 5. Build the source and target jobs for this profile.
 6. Submit the source job.
The source job should end with a return code of 0. If the source job does not complete successfully, you should examine the CKZERROR DD output from the source job to determine and resolve the problem. If you cannot resolve the problem, contact IBM Software Support.
 7. When the source job ends with a return code of 0, submit the target job.

What to do next

If both source and target jobs end with return code of 0, the installation has been verified.

Adding Db2 Cloning Tool to the Db2 Admin Launchpad

The Db2 Admin Launchpad is used to start installed IBM Db2 tools directly from a centralized panel. You can add Db2 Cloning Tool to the Db2 Admin Launchpad.

About this task

For instructions for adding Db2 Cloning Tool to the Db2 Admin Launchpad, go to [IBM Knowledge Center](#), find the online product documentation for your version of Db2 Administration Tool for z/OS, and search for "Db2 Admin Launchpad". The online product documentation describes how to create the Launchpad table and how to modify the Launchpad table to add a tool.

Exporting and importing ISPF cloning profiles and subsystem information

You can export cloning profiles from the Db2 Cloning Tool VSAM repository and Db2 subsystem information from the Db2 control file to external files by using the EXPORT command. After exporting to external files, you can import the cloning profiles and Db2 subsystem information to another or the same VSAM repository and Db2 control file by using IMPORT command.

If you use the export and import jobs to export from Db2 Cloning Tool V3.1 and import into Db2 Cloning Tool V3.2, fields that are new to V3.2 (such as **Use Group Attach**) will be populated with their defaults. After importing the profiles, you can modify the profiles in V3.2 to update the new field values.

You can omit export and import processing if you use the same VSAM repository and Db2 control file with Db2 subsystem information in Db2 Cloning Tool V3.2 as in Db2 Cloning Tool V3.1. Db2 Cloning Tool V3.2 automatically supports the processing of cloning profiles that were created in Db2 Cloning Tool V3.1.

Note: The export and import commands do not migrate table space cloning profiles to application cloning profile format; use the MIGRATE command as described in [“Migrating table space cloning profiles to application cloning profiles”](#) on page 103.

EXPORT command syntax

Sample JCL that includes EXPORT command syntax for exporting cloning profiles and Db2 subsystem information is located in the CKZEXUTL member of the SCKZJCL library.

Command

```
EXPORT
```

Required keywords

```
PROFILE-TYPE ( SS | TS )  
REPOSITORY-HLQ ( hlq )  
CONTROL-FILE-DSN ( dsname )  
{ PARM-DSN ( dsname )  
  CPARM-MEMBER ( member ) SPARM-MEMBER ( member ) |  
  CPARM-DDN ( ddname ) SPARM-DDN ( ddname ) }  
{ PROFILES ( "creator1"."name1", ... [ , "creatorn"."namen" ] ) |  
  PROFILE-LIKE ( "profilemask" ) CREATOR-LIKE ( "creatormask" ) }
```

Syntax

EXPORT

The EXPORT command is required.

PROFILE-TYPE (SS | TS)

Specifies the type of profiles that will be exported. Enter SS for subsystem cloning profiles or TS for table space cloning profiles.

REPOSITORY-HLQ (hlq)

Specifies the high level qualifier of the VSAM repository from which the profile data will be exported.

CONTROL-FILE-DSN (dsname)

Specifies the data set name of the Db2 control file from which the control file information will be exported.

PARM-DSN (dsname)

Required if a data set name and members are used to save exported data. It specifies the data set name in which to save exported profiles and subsystem data.

CPARM-MEMBER (member)

Required if a data set name and members are used to save exported data. It specifies the member name in which to save the exported cloning profile data.

SPARM-MEMBER (member)

Required if a data set name and members are used to save exported data. It specifies the member name in which to save the exported Db2 subsystem data.

CPARM-DDN (ddname)

Required if DD names are used to save exported data. It specifies the DD name for the exported cloning profile data.

SPARM-DDN (ddname)

Required if DD names are used to save exported data. It specifies the DD name for the exported Db2 subsystem data.

PROFILES ("creator1"."name1", ... [, "creatorn"."namen"])

Provides the list of profiles to be exported. Enter each cloning profile using the profile creator and profile name, separated by a period. The profile creator and profile name must each be surrounded by double quotes.

PROFILE-LIKE ("profile_mask")

Required if masks of the profile names are used to export profiles. An asterisk (*) is used to specify all profile names. You can use the wildcard characters (*) or (?) in any position of PROFILE-LIKE to limit the list of processed profiles. The profile name mask must be surrounded by double quotes.

CREATOR-LIKE ("creator_mask")

Required if masks of the creator names are used to export profiles. An asterisk (*) is used to specify all profile creators. You can use the wildcard characters (*) or (?) in any position of CREATOR-LIKE to limit the list of processed profiles. The profile creator mask must be surrounded by double quotes.

EXPORT syntax examples

Example 1

This example exports three table space cloning profiles from the DB2TOOLS.CKZ repository. The profiles and control file information will be exported to the CREATOR.CKZ.EXPORT data set. The cloning profile data will be exported to the CCPARM member and the control file information exported to the CSPARM member.

```
EXPORT
PROFILE-TYPE(TS)
REPOSITORY-HLQ(DB2TOOLS.CKZ)
CONTROL-FILE-DSN(DB2TOOLS.CKZ.DB2.CONTROL)
PARM-DSN(CREATOR.CKZ.EXPORT)
CPARM-MEMBER(CCPARM)
SPARM-MEMBER(CSPARM)
PROFILES("CREATOR"."PROFILE 01",
         "CREATOR"."PROFILE 02",
         "CREATOR"."PROFILE 03")
```

Example 2

This example exports all table space cloning profiles with creator names that begin with CR from the DB2TOOLS.CKZ repository. The profiles and control file information will be exported to the CREATOR.CKZ.EXPORT data set. The cloning profile data will be exported to the CCPARM member and the control file information exported to the CSPARM member.

```
EXPORT
PROFILE-TYPE(TS)
REPOSITORY-HLQ(DB2TOOLS.CKZ)
CONTROL-FILE-DSN(DB2TOOLS.CKZ.DB2.CONTROL)
PARM-DSN(CREATOR.CKZ.EXPORT)
CPARM-MEMBER(CCPARM)
SPARM-MEMBER(CSPARM)
PROFILE-LIKE("*")
CREATOR-LIKE("CR*")
```

IMPORT command syntax

Sample JCL that includes IMPORT command syntax for importing table space and subsystem cloning profiles and Db2 subsystem information from external files is located in the CKZIMUTL member of the SCKZJCL library.

Command

```
IMPORT
```

Required keywords

```
PROFILE-TYPE ( SS | TS )
REPOSITORY-HLQ ( hlq )
CONTROL-FILE-DSN ( dsname )
{ PARM-DSN ( dsname )
  CPARM-MEMBER ( member ) SPARM-MEMBER ( member ) |
  CPARM-DDN ( ddname ) SPARM-DDN ( ddname ) }
```

Optional keywords

```
[ IMPORT-ACTION-IF-PROFILE-EXISTS ( SKIP | REPLACE | NEW ) ]  
[ IMPORT-ACTION-IF-SUBSYSTEM-EXISTS ( SKIP | REPLACE ) ]  
[ USE-ORIGINAL-CREATORS ( Y | N ) ]
```

Syntax

IMPORT

The IMPORT command is required.

PROFILE-TYPE (*SS* | *TS*)

Specifies the type of profiles that will be imported. Enter SS for subsystem cloning profiles or TS for table space cloning profiles.

REPOSITORY-HLQ (*hlq*)

Specifies the high level qualifier of the VSAM repository that will contain the imported profiles.

CONTROL-FILE-DSN (*dsname*)

Specifies the data set name of the Db2 control file that will contain the imported control file data.

PARM-DSN (*dsname*)

Required if a data set name and members are used to import data. It specifies the data set name from which to import cloning profiles and subsystem data.

CPARM-MEMBER (*member*)

Required if a data set name and members are used import data. It specifies the member name from which to import cloning profiles.

SPARM-MEMBER (*member*)

Required if a data set name and members are used to import data. It specifies the member name from which to import Db2 subsystem data.

CPARM-DDN (*ddname*)

Required if DD names are used to import data. It specifies the DD name from which to import cloning profiles.

SPARM-DDN (*ddname*)

Required if DD names are used to import data. It specifies the DD name from which to import Db2 subsystem data.

[IMPORT-ACTION-IF-PROFILE-EXISTS (*SKIP* | *REPLACE* | *NEW*)]

(Optional) Specifies the action to take if a profile already exists in the VSAM repository. The following are valid values:

- SKIP (default): Skip the import of this profile.
- REPLACE: Replace the existing profile.
- NEW: Import the profile with a new generated name. After the utility is run, the log output contains CKZ6T112I messages that provide information about old and new profile names.

[IMPORT-ACTION-IF-SUBSYSTEM-EXISTS (*SKIP* | *REPLACE*)]

(Optional) Specifies the action to take if a Db2 subsystem already exists in the Db2 control file. The following are valid values:

- SKIP (default): Skip the import of this Db2 subsystem data.
- REPLACE: Replace the existing Db2 subsystem data.

[USE-ORIGINAL-CREATORS (*Y* | *N*)]

(Optional) Specifies whether to use the original profile creator instead of the user ID that runs the import command. The following are valid values:

- Y: (default) Use the original profile creator.
- N: Use the user ID that runs the IMPORT command as the profile creator.

IMPORT syntax example

This example imports table space cloning data into the DB2TOOLS . CKZ repository. The profiles and control file information will be imported from the CREATOR . CKZ . EXPORT data set. The cloning profile data will be imported from the CCPARM member and the control file information imported from the CSPARM member. If a profile exists in the repository, it will be imported and given a new name. If a Db2 subsystem exists in the repository, its information will be replaced with the imported subsystem data. All profiles will be named using the user ID that runs the IMPORT command as the profile creator.

```
IMPORT
PROFILE-TYPE(TS)
IMPORT-ACTION-IF-PROFILE-EXISTS(NEW)
IMPORT-ACTION-IF-SUBSYSTEM-EXISTS(REPLACE)
REPOSITORY-HLQ(DB2TOOLS.CKZ)
CONTROL-FILE-DSN(DB2TOOLS.CKZ.DB2.CONTROL)
PARM-DSN(CREATOR.CKZ.EXPORT)
CPARM-MEMBER(CCPARM)
SPARM-MEMBER(CSPARM)
USE-ORIGINAL-CREATORS(N)
```

Migrating table space cloning profiles to application cloning profiles

If you created table space cloning profiles under the previous version of the ISPF interface (also called the *classic ISPF interface*), you can migrate those profiles to the new application cloning profile format either by using the job that is provided with Tools Customizer or by creating a separate migration job.

After migration, the table space cloning profiles are preserved and can be accessed in the classic ISPF interface that is available from the new ISPF interface main menu.

By default, the Tools Customizer job migrates all table space cloning profiles to the new application cloning profiles in the same ISPF repository data set. For information about the Tools Customizer migration job, refer to the step to [Migrate ISPF repository profiles](#).

Alternatively, you can create a job that uses MIGRATE command syntax and additional keywords to limit the migration to a subset of table space cloning profiles, to simulate the migration, and to send the migrated application cloning profiles to a different repository.

Command

```
MIGRATE
```

Required DD

The CKZSLIB DD must be specified and must point to the Db2 Cloning Tool skeleton library.

Required keywords

```
SOURCE-REPOSITORY-HLQ ( source-hlq )
TARGET-REPOSITORY-HLQ ( target-hlq )
```

Optional keywords

```
PROFILE-TYPE ( ALL | SS | TS )
PROFILE-LIKE ( "*" | "profile_mask" )
CREATOR-LIKE ( "*" | "creator_mask" )
MIGRATE-ACTION-IF-PROFILE-EXISTS ( SKIP | REPLACE | NEW )
SIMULATE ( Y | N )
```

Syntax

MIGRATE

The MIGRATE command is required.

SOURCE-REPOSITORY-HLQ (*source-hlq*)

Specifies the high level qualifier of the VSAM repository that contain the original profiles.

TARGET-REPOSITORY-HLQ (*source-hlq*)

Specifies the high level qualifier of the VSAM repository that will contain the migrated profiles.

PROFILE-TYPE (**ALL** | **SS** | **TS**)

(Optional) Specifies the type of profiles that will be migrated. Enter SS for subsystem cloning profiles, enter TS for table space cloning profiles, or enter ALL for both types of profiles.

MIGRATE-ACTION-IF-PROFILE-EXISTS (**SKIP** | **REPLACE** | **NEW**)

(Optional) Specifies the action to take if a profile already exists in the target VSAM repository. The following are valid values:

- SKIP (default): Skip the migration of this profile.
- REPLACE: Replace the existing profile.
- NEW: Migrate the profile with a new generated name. After the utility is run, the log output contains messages that provide information about old and new profile names.

SIMULATE (**Y** | **N**)

(Optional) Specifies whether simulate mode is enabled. If simulate mode is enabled, no changes will be added into the target VSAM repository.

MIGRATE examples

Example 1: Syntax only

This example migrates all cloning profiles from the DB2TOOLS.CKZ32 repository into the DB2TOOLS.CKZ41 repository. If a profile exists in the new repository, it is skipped.

```
MIGRATE
SOURCE-REPOSITORY-HLQ (DB2TOOLS.CKZ32)
TARGET-REPOSITORY-HLQ (DB2TOOLS.CKZ41)
```

Example 2: Syntax and JCL

This JCL example simulates the migration of all table space cloning profiles with creator like "TSUSER" and name like "*" to application cloning profiles.

```
//MIGRATE JOB ,TSUSER,CLASS=A,MSGCLASS=X,REGION=0M,NOTIFY=&SYSUID
//*
//STEP1 EXEC PGM=CKZ8UTIL
//STEPLIB DD DISP=SHR,DSN=RSQA.CKZ320.IBMTAPE.SCKZLOAD
//CKZPRINT DD SYSOUT=*
//CKZERROR DD SYSOUT=*
//CKZSLIB DD DISP=SHR,DSN=RSQA.CKZ320.IBMTAPE.SCKZSLIB
//SYSIN DD *
MIGRATE
SOURCE-REPOSITORY-HLQ(DB2TOOLS.CKZ32)
TARGET-REPOSITORY-HLQ(DB2TOOLS.CKZ41)
PROFILE-TYPE(TS)
MIGRATE-ACTION-IF-PROFILE-EXISTS(NEW)
SIMULATE(Y)
PROFILE-LIKE("*")
CREATOR-LIKE("TSUSER")
```

Note: Application cloning profiles cannot be migrated to table space cloning profiles.

Db2 version migration considerations

Once Db2 Cloning Tool has been installed and configured, follow these steps if a Db2 subsystem being used with Db2 Cloning Tool is migrated to a later version or must fallback to a previous version.

Db2 Cloning Tool Subsystem Cloning

When migrating from one Db2 version to a later version, a rebind on the new subsystem is required for the plans and packages that were used by Db2 Cloning Tool Subsystem Cloning on Db2 subsystems. Rebind the plans and packages on each subsystem on which Db2 Cloning Tool Subsystem Cloning will be used. In general, the steps are:

1. Start Tools Customizer and specify to customize Db2 Cloning Tool.
2. On the **Customizer Workplace** panel, edit the product parameters.
3. On the **Product Parameters** panel, select the **Run Subsystem Cloning package binds** task. Save and exit.
4. Create and associate the new Db2 subsystem.
5. On the **Customizer Workplace** panel, edit the new subsystem.
6. On the **DB2 Parameters** panel, specify the mode, level, and other required Db2 subsystem parameters. Be sure to verify or specify the Db2 APPLCOMPAT bind option. Save and exit.
7. Generate the jobs for the subsystem.

Refer to [“Roadmap: Recustomizing Db2 Cloning Tool” on page 79](#) for additional information.

Db2 Cloning Tool Table Space Cloning

When migrating from one Db2 version to a later version, a rebind on the new subsystem is required for the plans and packages that were used by Db2 Cloning Tool Table Space Cloning on Db2 subsystems. Rebind the plans and packages on each subsystem on which Db2 Cloning Tool Table Space Cloning will be used. In general, the steps are:

1. Start Tools Customizer and specify to customize Db2 Cloning Tool.
2. On the **Customizer Workplace** panel, edit the product parameters.
3. On the **Product Parameters** panel, select the **Run Table Space Cloning package binds** task. Save and exit.
4. Create and associate the new Db2 subsystem.
5. On the **Customizer Workplace** panel, edit the new subsystem.
6. On the **DB2 Parameters** panel, specify the mode, level, and other required Db2 subsystem parameters. Be sure to verify or specify the Db2 APPLCOMPAT bind option. Save and exit.
7. Generate the jobs for the subsystem.

For more information, refer to [“Roadmap: Recustomizing Db2 Cloning Tool” on page 79](#).

Fallback instructions

If after migrating to a later Db2 version, you must fall back to the prior version of Db2, do the following:

- For Db2 Cloning Tool Subsystem Cloning, the plans and packages must be rebound on each subsystem on which Db2 Cloning Tool Subsystem Cloning will be used. In general, the steps are:
 1. Start Tools Customizer and specify to customize Db2 Cloning Tool.
 2. On the **Customizer Workplace** panel, edit the product parameters.
 3. On the **Product Parameters** panel, select the **Run Subsystem Cloning package binds** task. Save and exit.
 4. If not already done, create or associate the prior version Db2 subsystem.

5. On the **Customizer Workplace** panel, edit the prior version subsystem.
6. On the **DB2 Parameters** panel, specify the mode, level, and other required Db2 subsystem parameters. Be sure to verify or specify the Db2 APPLCOMPAT bind option. Save and exit.
7. Generate the jobs for the subsystem.

For more information, refer to [“Roadmap: Recustomizing Db2 Cloning Tool” on page 79](#).

- For Db2 Cloning Tool Table Space Cloning, the plan and packages must be rebound on each subsystem on which Db2 Cloning Tool Table Space Cloning will be used. In general, the steps are:
 1. Start Tools Customizer and specify to customize Db2 Cloning Tool.
 2. On the **Customizer Workplace** panel, edit the product parameters.
 3. On the **Product Parameters** panel, select the **Run Table Space Cloning package binds** task. Save and exit.
 4. If not already done, create and/or associate the prior version Db2 subsystem.
 5. On the **Customizer Workplace** panel, edit the prior version subsystem.
 6. On the **DB2 Parameters** panel, specify the mode, level, and other required Db2 subsystem parameters. Be sure to verify or specify the Db2 APPLCOMPAT bind option. Save and exit.
 7. Generate the jobs for the subsystem.

For more information, refer to [“Roadmap: Recustomizing Db2 Cloning Tool” on page 79](#).

Chapter 6. Planning for copying and renaming volumes

Before attempting to actually use the Db2 Cloning Tool process, some planning and decision-making should take place. This topic discusses those things that need to be considered to clone a Db2 subsystem or a volume.

Selection of source and target volumes

The following requirements should be considered when planning the cloning environment.

Scope of source volumes to be copied

The source volumes copied must include all data sets required by the application that will access the renamed data sets on the target volumes. It is recommended that the source Db2 SDSNLOAD and SDSNEXIT libraries on source volumes should not be cloned into the target Db2 SDSNLOAD and SDSNEXIT libraries. The reasons for this recommendation are:

- There may be problems setting up APF authorization of the target SDSNLOAD and SDSNEXIT libraries if the target volumes are not SMS managed and Db2 Cloning Tool is used to do the copy. The target volumes these libraries get cloned to might be different for different runs.
- Member DSNHDECP has the default Db2 SSID in it. This module can only have one name and the default SSID should be different between the source and target Db2.

The Db2 directory, Db2 catalog, active logs, and BSDSs must be contained on the source volumes, as well as the application databases.

Desired target volume data sets

Any copies of data sets to be used by the application must reside within the list of "cloned" volumes, and must match a rename mask. Db2 Cloning Tool does not require all data sets on target volumes be renamed. The NOTRENAMED parameter is used to specify:

- The disposition of any data sets not matched to a rename mask.
- The return code if at least one data set is not renamed.

If all data sets on all copied volumes are critical, a return code of eight (8) should be requested, as an indication that the errors must be reviewed and the process rerun. This assumes that a return code of eight (8) is treated by the application as a RENAME failure.

If rename masks intentionally match only some data sets, specify a return code of zero (0) or four (4). Normally a disposition of DELETE should be specified if the volumes are SMS managed (in keeping with SMS rules), and/or if the space occupied by not-renamed data sets may be needed for subsequent allocations.

Note: By not renaming all data sets, the process is essentially a selective data set copy, with the caveat that those data sets renamed and kept occupy the same track locations as their source volume counterparts.

Db2 Cloning Tool deletes, on the target volumes, any temporary data sets created from the volume copies. Db2 Cloning Tool does *not* detect, catalog, or delete any data sets on target volumes whose source volume counterparts are uncataloged. Despite not being cataloged, the target volume data set will be renamed, if it matches a rename mask.

Note: For performance reasons, Db2 Cloning Tool does not diagnose from volume to BCS to detect anomalies, such as uncataloged data sets, especially when the anomaly can be replicated each cycle of

the process. It is recommended that volumes and catalogs be diagnosed routinely to identify and repair the source of any problems.

An ICF catalog can be renamed, but it will not be usable as an ICF catalog.

Source and target volume condition

To prevent volume internal VTOC index and VVDS errors, the VTOC, VTOCIX, and VVDS should be in the same location on the target volumes as they are on the source volumes prior to the COPY taking place.

Target volume online status

Db2 Cloning Tool expects that the target volumes will be online to only one system while RENAME runs. If the target volumes are online to more than one system, VTOC index and VVDS errors may occur.

Data set/sphere integrity

By default, Db2 Cloning Tool will detect and fail the process if one of the data set integrity violations listed in the following table is encountered:

| <i>Table 31. Data set/sphere integrity violations</i> | |
|---|---|
| Integrity violation | Description |
| Multivolume data sets | All segments of multivolume data sets, where the source volume data set matches a rename mask, must be wholly contained within the list of source volumes. When we use the term 'multivolume data sets', multivolume data sets, striped data sets, and VSAM spheres are implied |
| VSAM spheres | All components and associations of VSAM spheres, where the cluster name matches a rename mask, cannot be partially on the source volumes. This means the volume list must include all components of a cluster and any alternate indexes and their components that are associated with the base. |
| GDGs | All active generations of a GDG base, where the base name matches a rename mask, cannot be partially on the source volumes. Exceptions are allowed for active generations that are on tape or have been migrated, but references to such target entries will fail. |
| Non-VSAM data set aliases | If a data set found on a volume matches a rename mask, and the data set has one or more aliases, all aliases must also match some rename mask. |

Discrete volumes vs. SMS storage group specification

Specification of volumes by SMS storage groups (available only for FlashCopy, SnapShot, or TimeFinder/Clone) will reduce maintenance required to ensure that volumes used by an application are included in the FlashCopy, SnapShot, or TimeFinder/Clone list. If storage group specification is used, exclude options are available to exclude source or target volumes from the storage group list.

For example, there is a source SMS storage group SRCSG1 and a target SMS storage group TGTSG1. SMS storage group SRCSG1 has two volumes: SRC001 and SRC002. SMS storage group TGTSG1 has two volumes: TGT001 and TGT002. The COPY is using FROM-STORAGEGROUP(SRCSG1) and TO-STORAGEGROUP(TGTSG1). COPY would use as source volumes SRC001 and SRC002, and would use TGT001 and TGT002 as target volumes.

Now volume SRCP01 is added to SRCSG1 and volume TGTP01 is added to TGTSG1. Now COPY would use as source volumes SRC001, SRC002, and SRCP01, and would use TGT001, TGT002, and TGTP01 as target volumes. By using SMS storage groups, the COPY statements do not have to be changed when volumes are added or removed from the SMS storage groups. COPY will use whatever volumes are currently defined in the specified SMS storage groups.

Migrated application data sets

If Db2 subsystems contain generation data sets, users must ensure that data sets to be copied are not allowed to migrate before the copy volumes are created. This includes any active generations of a renamed GDG.

If you COPY source volumes, RENAME data sets on the new target volumes, update the Db2 internals, and then start processing against the target volumes, your processing will fail if a job tries to access a renamed generation that was migrated at the time of COPY, and therefore not copied. Your entire process could then be compromised. Db2 Cloning Tool does NOT copy migrated data.

Requirements for source and target volume pairing

Db2 subsystem use of source volumes must be planned such that enough target volumes exist to allow all source volumes to be paired with a target volume.

FlashCopy source and target volumes:

- must be in the same subsystem
- must have the same track format (e.g., 3380 vs. 3390)
- the target volume size must be equal to or greater than the source volume

For more information on FlashCopy source and target pairing rules, contact IBM.

SnapShot source and target volumes:

- must be in the same RVA partition
- must have the same track format (e.g., 3380 vs. 3390)
- the target volume size must be equal to or greater than the source volume

For more information on SnapShot source and target pairing rules, contact StorageTek.

EMC TimeFinder/Clone source and target volumes:

- must be in the same subsystem
- must have the same track format (e.g., 3380 vs. 3390)
- the target size must be equal to or greater than the source volume

For more information on TimeFinder/Clone source and target pairing rules, contact EMC.

Volume relationship conflicts

Care should be taken to ensure that when Db2 Cloning Tool initiates FlashCopy, SnapShot, or TimeFinder/Clone, no other relationships exist that would cause Db2 Cloning Tool to reject a volume, or DSS or EMCSNAP to fail.

If other relationships are desired after the Db2 Cloning Tool process is complete, a COPYCHECK command is provided to wait for all volume copies to complete.

Ensure RACF and ACF authorities

Before cloning, you should ensure that all proper RACF and ACF permissions and privileges are in place. Refer to [“Verify that your environment meets security requirements”](#) on page 22 for information about required permissions and privileges.

Data set renaming considerations

Existing naming conventions will dictate the required rename masks to ensure that all required data sets are renamed, and that renaming does not result in two or more data sets renamed to the same target name. Because Db2 Cloning Tool renaming allows introduction of additional qualifiers for the logs and

BSDSs, users may wish to consider a data set name length restriction of something less than 44 bytes (35 for GDG base names).

If the RENAME masks cause the data set names to be longer than the source volume data set names, and/or the target ICF catalog name is longer than the source ICF catalog name, the VVDS on the source volume must be large enough to support the expanded target names. Db2 Cloning Tool does not add extents to target VVDS data sets.

Because of Db2 naming standard requirements, only the Db2 logs and BSDSs can be renamed beyond the *hlq*. All other Db2 files can only be renamed at the *hlq*:

- Db2 directory – Db2 expects a specific naming standard:

```
hlq.DSNDBx.DSNDB01.*
```

- Db2 catalog – Db2 expects a specific naming standard:

```
hlq.DSNDBx.DSNDB06.*
```

- Db2 databases – Db2 expects a specific naming standard:

```
vcat.DSNDBx.dbname.psname.y0001.Annn
```

or

```
vcat.DSNDBx.dbname.psname.y0002.Annn
```

Target ICF catalog considerations

Db2 Cloning Tool will catalog target volume data sets to either a populated or an empty ICF catalog. If a target catalog entry already exists, the RECATALOG option of the RENAME command is required. However, the BCSCLEAN command can be used to ensure this doesn't happen.

Cataloging to a populated ICF catalog may extend the execution time. When RECATALOG is specified, extra care should be taken, both initially and when maintaining the rename masks, to ensure that a target catalog entry is not replaced as the result of an incorrect target rename mask. If there are no compelling reasons for the target catalog to hold data sets other than those resulting from a Db2 Cloning Tool process, a discrete catalog for renamed data sets is advised.

As noted in the description of the RECATALOG keyword, using the BCSCLEAN command allows target catalogs to be used by applications other than the one(s) involved with the copy process, without using the RECATALOG option. For more information, see the [“BCSCLEAN” on page 379](#) topic.

The Db2 Cloning Tool journal data set should not be cataloged in the target ICF catalog. This can cause the Db2 Cloning Tool RENAME command to deadlock with itself over access to the target ICF catalog.

Location of the source and target ICF catalogs

Source ICF catalog information for source volume data sets being cloned is required at the point-in-time by Db2 Cloning Tool. This requirement enables the rename of the target volume data sets. Some information like VSAM sphere and GDG Base information is in the ICF catalogs only, and not on the DASD being cloned and therefore, needs to be captured at the point-in-time. Two options are available: either the source ICF catalogs can reside on source volumes being cloned, and be read from the target volumes; or the source ICF catalogs can be on non-source volumes and be read from the live source ICF catalogs at the time the replication or split occurs.

Target ICF catalogs used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

Target ICF catalogs that are used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step, unless the

TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword of the COPY command is specified. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

If the target ICF catalog was on the source volume when it was cloned, it can be renamed as part of the cloning process just like any other data set, but it will not be usable as a catalog.

This example uses a cloning of one source volume to one target volume that has one source ICF catalog and one target ICF catalog, where:

- SRC001 is the source volume
- TGT001 is the target volume
- VOL001 is a volume that is not being cloned
- SRC.CATALOG is the source ICF catalog
- TGT.CATALOG is the target ICF catalog

SRC.CATALOG can be on volumes SRC001 or VOL001, but cannot be on volume TGT001. If SRC.CATALOG is on SRC001, its copy on TGT001 can be renamed, but cannot be used.

TGT.CATALOG can be on volumes SRC001 or VOL001 but cannot be on volume TGT001 from volume copy through RENAME. If it is desired to have TGT.CATALOG be on TGT001, then it must be moved to some other volume before the volume copy, and then can be moved back to TGT001 after RENAME has completed. If it is on SRC001, its copy on TGT001 can be renamed, but cannot be used.

Target data set ICF catalog aliases

Users are responsible for creating ICF catalog aliases, if needed for new target data set names.

This must be coordinated with the rename masks used, and it must be determined whether the MLA setting is different for the image(s) from where source volume data sets are accessed, compared to the MLA setting for the image(s) from where target volume data sets are to be accessed.

Note: To determine the target ICF catalogs you may wish to create, as well as the aliases needed for target volume data sets, see the FINDUCATS command. FINDUCATS identifies catalogs involved with source volume data sets and the aliases employed for source volume data sets.

Considerations for DFSMS pervasive encryption

If source volumes contain data sets that are encrypted with the DFSMS pervasive encryption, consider the following points.

- For cross-LPAR cloning, any key labels that are used to protect data sets on source volumes must be defined on the target LPAR and must refer to the same encryption key.
- The target Db2 started task user ID can use any key labels that are used to protect Db2 data sets that are cloned from the source Db2 subsystem.

Return code choices

A number of Db2 Cloning Tool options allow the user a choice of return codes if at least one circumstance is encountered for the keyword involved.

In other words, the user may choose the seriousness of a situation. Scenarios discussed in this document assume that the conditional execution of subsequent steps adheres to the convention that return code 0 means successful, 4 means warning, and 8 means an error.

The following is an example of choosing to treat a RENAME not-renamed situation as a critical error (only partial JCL and control statements included):

```
//RENAME EXEC PGM=CKZ00010
//CKZIN DD *
RENAME -
RENAME-MASKS(PROD.** TEST.**) -
```

```

NOTRENAMED(DELETE,RC(8)) -
JOURNAL-DDN(JOURNAL)
//IFBAD IF (RENAME.RC >= 8) THEN
//TGTBAD EXEC PGM=program-to-signal-rename-failed
//
// ELSE
//TARGETOK EXEC PGM=application-program-that-uses-target-volumes
//IFBAD ENDIF

```

Cloning a Db2 subsystem

To clone a Db2 subsystem, the target subsystem must be set up.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: When Db2 terminates, the MSTR, IRLM, DBM1, and DIST address spaces are marked as unavailable. Each Db2 termination reduces the number of available address spaces. The system must be IPLed to make those address spaces available again.

The following requirements also must be met:

1. The Db2 directory, Db2 catalog, active logs, and BSDSs must be contained on the source volumes that are being cloned, as well as the application databases.
2. The source Db2 SDSNLOAD and SDSNEXIT libraries on source volumes should not be cloned into the target Db2 SDSNLOAD and SDSNEXIT libraries. The reasons for this recommendation are:
 - There might be problems setting up APF authorization of the target SDSNLOAD and SDSNEXIT libraries if the target volumes are not SMS managed and Db2 Cloning Tool COPY is used to do the copy. The target volumes these libraries get cloned to might be different for different runs.
 - Member DSNHDECP has the default Db2 SSID in it. This module can have only one name and the default SSID should be different between the source and target Db2.
3. Because of Db2 naming standard requirements, only the Db2 logs and BSDSs can be renamed beyond the high-level qualifier. All other Db2 files that are shown in the list that follows can be renamed only at the high-level qualifier:
 - Db2 directory – Db2 expects a specific naming standard:

```
hlq.DSNDBx.DSNDB01.*
```

- Db2 catalog – Db2 expects a specific naming standard:

```
hlq.DSNDBx.DSNDB06.*
```

- Db2 databases – Db2 expects a specific naming standard:

```
vcat.DSNDBx.dbname.psname.y0001.Annn
```

The target Db2 SDSNEXIT can be set up before the Db2 Cloning Tool processing is started. It should be an APF authorized library.

It is recommended that the target Db2 system be run with the same Db2 release and the same or similar maintenance level as the source Db2 system. The reason for this recommendation is that a different release or maintenance level of Db2 might have dependencies on the Db2 catalog, directory, or BSDS that will not exist as part of the cloning. After the cloning is complete, the target Db2 could then be migrated to a higher release or maintenance level of Db2.

The following steps can be set up before the Db2 Cloning Tool processing is started. Normally, these will be a "one-time" setup and do not have to be repeated.

1. Set up normal zparm, DSNZPARx, for the target Db2 subsystem.

Starting with a copy of the source system's zparms would be desirable. Check the macro keywords for items that might need to be changed to reflect the target subsystem. For example, CATALOG=, IRLMPRC=, and IRLMSID= would need to be changed for the target subsystem. Assemble and link-edit DSNZPARx to the target Db2 SDSNEXIT LOAD library.

2. Set up DSNHDECP for the target subsystem. The target DSNHDECP should be the same as used by the source system.
3. Assemble and link-edit DSNHDECP to the target Db2 SDSNEXIT LOAD library.
4. Set up the required started tasks JCL for your target subsystem: DSN?MSTR, DSN?DBM1, DSN?DIST, and so on.
5. Set up the target subsystem Db2 and target subsystem IRLM subsystem names in SYS1.PARMLIB, IEFSSNnn.

You may use the SETSSI commands to add the target subsystem names to avoid an IPL, but ensure that the PARMLIB member is updated.

6. Set up special zparm, DSNZSPEC, for the target Db2 subsystem.

This special zparm will allow the target's Db2 catalog to be updated and defers the backout of in-flight transactions on the target subsystem. This zparm should only be used for the time needed to update the target's VCATNAMEs and, optionally, the target's Db2 storage group names.

- a. Allocate a special macro library for DSNZSPEC. It will be a small PDS with only one member, DSN6SPRC.
- b. Copy member DSN6SPRC from the distributed SDSNMACS library to the special macro library.
- c. Change special macro library member, DSN6SPRC, as follows:

Change this:

```
&SPRMCTU SETC '0' YES=CATALOG CAN BE UPDATED
```

To this:

```
&SPRMCTU SETC '1' YES=CATALOG CAN BE UPDATED
```

- d. Save the modified special macro library member. For example, the modified special DSN6SPRC macro might look something like (in part):

```
...
&SPRMNAP SETC '0'          BIT ON - SKIP ADJ. PREFETCH @KYF1570
&SPRMSHP SETC '0'          BIT ON - SIMULATE 2G HIPERSPACE
&SPRMCTU SETC '1'          YES=CATALOG CAN BE UPDATED
&SPRMXPL SETC '0'          YES=GEN ALL EXPLAIN TABLES
&SPRMNHJ SETC '0'          YES=TURN OFF HYBRID JOIN
...
```

- e. Create DSNZSPEC.

This should be copied from the normal zparm, DSNZPARx, created for the target Db2 subsystem.

- Change DSNZSPEC macro DSN6SPRM from RESTART, ALL to DEFER, ALL.
- Change DSNZSPEC macro DSN6SPRM keyword SYSADM or SYSADM2 to specify the user ID that will be running the SQL statements, described in Step 7, on the target Db2.
- Change the JCL for DSNZSPEC so that the special macro library is the first library in the assembly step //SYSLIB DD concatenation.
- Change all occurrences of DSNZPARx to DSNZSPEC, except on the link-edit card INCLUDE ADSNLOAD (DSNZPARG). Assemble and link-edit DSNZSPEC to the target Db2 SDSNEXIT LOAD library.

For example, the DSNZSPEC might look something like (in part):

```
...
DSN6ENV      MVS=XA
DSN6SPRM     DEFER,
              ALL,
              X
              X
...
```

```
          SYSADM=CKZUSER,           X
          SYSADM2=DB2ADM,          X
...
//SYSLMOD DD DISP=SHR,
//      DSN=target.SDSNEXIT
...
  ENTRY   DSNZMSTR
  NAME    DSNZSPEC(R)
...
```

7. If not already done, bind the Db2 Cloning Tool plan and package on the source subsystem.

To update the Db2 catalog on the target Db2 subsystem, the Db2 Cloning Tool **DB2SQL** and **DB2SCHEMA-UPDATE** commands have a plan and package that need to be installed. If the binds for the plan and package have already been completed on the source Db2 subsystem, they will be available for use on the target Db2 subsystem when needed without having to explicitly bind them on the target Db2 subsystem after the cloning. If you choose *not* to bind the plan and package on the source subsystem, you will need to bind them on the target subsystem after cloning.

You can use Tools Customizer to generate and submit the bind jobs. If you are planning to use the stored procedure, ensure the binds related to the stored procedure are done. See the topic [Chapter 4, “Customizing Db2 Cloning Tool,”](#) on page 77.

Chapter 7. Setting up to copy by volume with FlashCopy, SnapShot, or TimeFinder/Clone

Follow this procedure if volume copies are to be created with volume FlashCopy or SnapShot via DFSMSdss, or TimeFinder/Clone.

For volume copies created in other ways, see [Chapter 8, “Setting up to copy by volume with an onsite mirror tool,”](#) on page 119.

The steps to accomplish the Db2 Cloning Tool function are placed within the application's job stream, typically in multiple locations, in order for the function steps to run at the appropriate times, relative to the application's activities. The Db2 Cloning Tool steps are invoked via commands to a common program.

If multiple commands are supplied within the same step, any command that results in a return code equal to or greater than eight will terminate the step. For instance, if COPY and RENAME are in the same step, and COPY terminates with a return code 8, because insufficient target volumes were found, RENAME will be bypassed.

Several commands specific to cloning a Db2 subsystem are not mentioned here. See [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 for more information.

The available commands are:

FINDUCATS

Locates ICF user catalogs pointing to source volume data sets.

COPY

Initiates copies and captures ICF catalog data.

RENAME

Renames and catalogs target volume data sets.

BCSCLEAN

Deletes catalog entries from previous executions.

Db2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone

The table that follows summarizes the steps required to create volume copies with volume FlashCopy, SnapShot, or TimeFinder/Clone.

| Step | Description |
|-------------|--|
| 1 | FINDUCATS step. Optional step to run one time or occasionally, to find ICF user catalogs pointing to source volume data sets. |
| 2 | Shut down or suspend the source Db2 subsystem. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for additional instructions. These instructions are a prerequisite to Step 3. This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone. |
| 3 | COPY step. |
| 4 | Start up or resume the source Db2 subsystem. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for additional instructions. These instructions are a prerequisite to Step 5. This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone. |
| 5 | RENAME step. This step renames and catalogs target volume data sets. |

Table 32. Db2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone (continued)

| Step | Description |
|------|---|
| 6 | Db2 cloning procedures. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for additional instructions. |
| 7 | BCSCLEAN step. Optional step runs after target volume usage is terminated and before the next COPY. |

Steps for volume FlashCopy, SnapShot, or TimeFinder/Clone

These are the steps to create volume copies with volume FlashCopy, SnapShot, or TimeFinder/Clone.

Procedure

1. FINDUCATS Step (optional).

FINDUCATS identifies which ICF User catalogs point at data sets on the source volumes to be copied.

The COPY step requires pairs of source/target user ICF catalogs to be specified. FINDUCATS does not negate this need. It is intended to be run prior to initial setup, and possibly on an occasional basis, to make sure the user catalogs that should be specified for the COPY step have not changed.

2. Shut down or suspend the source Db2 subsystem.

- If an offline clone is desired (Db2 is shut down), choose one of the offline cloning procedures in [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 and complete that procedure prior to beginning Step 3.
- If an online clone is desired (Db2 is suspended), choose one of the online cloning procedures in [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 and complete that procedure prior to beginning Step 3.

Note: If you are cloning with consistent FlashCopy, SnapShot, or TimeFinder/Clone, you do not need to suspend the source Db2 subsystem.

These instructions are a prerequisite to Step 3.

3. COPY step.

The COPY step initiates volume copies, and concurrently backs up the source ICF catalog data relevant to the data sets on the source volumes being copied. The COPY step completes when copy initiations are complete and all ICF catalog data has been backed up. Source ICF catalogs can also be cloned and read from the target volume. See the COPY command for more information.

4. Start or resume the source Db2 subsystem.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone.

Access, including modification to source volumes, may be resumed after the COPY step completes successfully. However, if time is not of the essence regarding source volume access, or if the RENAME step runs in a relatively short time frame, it is recommended that source volume access resume only after the RENAME step has completed.

Errors such as volumes not specified that should have been, or catalogs incorrectly specified, are not detected until the RENAME step. These categories of errors require the COPY step to be rerun. If the source volumes have changed since the first running of the COPY step, the same P.I.T. (point-in-time) images cannot be captured.

A typical implementation of the COPY step is to use it as a trigger for an application requiring access to source volumes. This can be accomplished by positioning the COPY job as a predecessor to the application, via a job scheduler.

- If an offline clone was done, start the source Db2 subsystem up. See [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125.

- If an online clone was done, resume the source Db2 subsystem. See [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125.

5. **RENAME step.**

The RENAME step renames and catalogs target volume data sets. The SAFE option, a keyword of the RENAME command is recommended so the RENAME step can be rerun to correct problems caused by mistakes such as incorrectly specified rename masks.

Because of Db2 naming standard requirements, only the Db2 logs and BSDSs can be renamed beyond the hlq. All other Db2 files can only be renamed at the hlq:

- Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB01.**
- Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB06.**
- Db2 databases – Db2 expects a specific naming standard
vcat.DSNDBx.dbname.psname.y0001.Annn

6. **Db2 cloning procedures.**

Perform one of the cloning procedures found in [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 before beginning Step 7.

7. **BCSCLEAN step (optional).**

Assuming the Db2 subsystem clone is repetitive, each cycle of the process will likely leave orphaned catalog entries for data sets not used in subsequent cycles of the application.

If the target catalog(s) is dedicated to just the target volume data sets, this problem can be avoided by simply placing an IDCAMS step before the COPY step to delete and redefine the target catalog(s) or use BCSCLEAN to delete ICF catalog entries for data sets Db2 Cloning Tool knows were created during the last cycle. If the redefined ICF catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the ICF catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the ICF catalog.

If the target ICF catalog is used for data sets other than those on target volumes, the BCSCLEAN function will delete ICF catalog entries for data sets Db2 Cloning Tool knows were created during the last cycle. This step can be placed anywhere between the end of target volume access and the next start of the COPY process.

Chapter 8. Setting up to copy by volume with an onsite mirror tool

Follow this procedure if clones are created by an onsite mirror tool that does an establish, waits for the data to be copied to the target volume, and then splits the mirror relationship at the appropriate point in time. This could include user-created clones and clones created with IBM PPRC, EMC TimeFinder/Mirror, Hitachi ShadowImage, Softek Replicator, and Innovation Data Processing FDRPAS.

If volume copies are to be created with FlashCopy or SnapShot via DFSMSdss, or TimeFinder/Clone, see Chapter 7, “Setting up to copy by volume with FlashCopy, SnapShot, or TimeFinder/Clone,” on page 115.

The steps to accomplish the Db2 Cloning Tool function are placed within the application's job stream, typically in multiple locations, in order for the function steps to run at the appropriate times, relative to the application's activities. The Db2 Cloning Tool steps are invoked via commands to a common program.

If multiple commands are supplied within the same step, any command that results in a return code equal to or greater than eight will terminate the step. For instance, if COPY and RENAME are in the same step, and COPY terminates with a return code 8, because insufficient target volumes were found, RENAME will be bypassed. Several commands specific to cloning a DB subsystem are not mentioned here. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for more information.

The available commands are:

FINDUCATS

Locates ICF user catalogs pointing to source volume data sets.

COPY

Captures ICF catalog data, and optionally, clips and varies target volumes online.

RENAME

Renames and catalogs target volume data sets.

BCSCLEAN

Deletes catalog entries from previous executions.

Db2 Cloning Tool process for user-created target volume clones

The table that follows summarizes the steps required for user-created clones.

| Step | Description |
|-------------|---|
| 1 | FINDUCATS step. Optional step to run one time or occasionally, to find user catalogs pointing to source volume data sets. |
| 2 | User-supplied step to establish continual mirrors. |
| 3 | Shut down or suspend the source Db2 subsystem. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for additional instructions. These instructions are a prerequisite to Step 4. This step is not necessary if using consistent split or break mirror. |
| 4 | User-supplied step to split or break mirrors. |
| 5 | COPY step. This step backs up the source ICF catalogs that point to the source volume data sets in synchronization with the split. Source ICF catalogs can also be cloned and read off the target volumes. See the COPY command for more information. |
| 6 | Start up or resume the source Db2 subsystem. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for additional instructions. These instructions are a prerequisite to Step 7. This step is not necessary if using consistent split or break mirror. |

Table 33. Db2 Cloning Tool process for user-created target volume clones (continued)

| Step | Description |
|------|--|
| 7 | RENAME step. This step renames and catalogs target volume data sets. |
| 8 | Db2 Cloning procedure. See Chapter 9, “Cloning Db2 subsystems,” on page 125 for additional instructions. |
| 9 | BCSCLEAN step. Optional step runs after target volume usage is terminated and before the next Db2 Cloning Tool COPY. |

Steps for user-created target volumes

These are the steps user-created target volumes.

Procedure

1. FINDUCATS step (optional).

FINDUCATS identifies which user ICF catalogs contain entries for data sets on the source volumes to be copied. The COPY step requires pairs of source/target user ICF catalogs to be specified. FINDUCATS does not negate this need. It is intended to be run prior to initial setup, and possibly on an occasional basis, to make sure the user catalogs that should be specified for the COPY step have not changed.

2. User-supplied step to establish continual mirrors.

This step must be supplied by the user and placed in the application stream, sufficiently ahead of the desired 'split time', such that the target volumes will be synchronized with their source volume counterparts.

3. Shut down or suspend the source Db2 subsystem.

- If an offline clone is desired (Db2 is shut down), choose one of the offline cloning procedures in [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 and complete that procedure prior to beginning Step 4.
- If an online clone is desired (Db2 is suspended), choose one of the online cloning procedures in [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 and complete that procedure prior to beginning Step 4.

Note: If you are cloning with consistent split or break mirror, you do not need to suspend the source Db2 subsystem.

These instructions are a prerequisite to Step 4.

4. User-supplied step to split or break mirrors.

This step must be supplied by the user to split mirrors created in step 2. For Hitachi ShadowImage, when using the ICKDSF PPRCOPY commands, the pairs must be suspended with a steady split request before deleting the pairs. The suspend with steady split request causes any pending updates to the target volume to be externalized.

5. COPY step.

The COPY step in this scenario is issued with the 'DATA-MOVER(PGM(NONE))' parameter. 'DATA-MOVER(PGM(NONE))' implies that target volumes have already been created. The COPY command in this case only backs up the source ICF catalog information needed to rename and catalog the target volume data sets. Optionally, it relabels the target volumes and varies them online for RENAME processing. The COPY should be done immediately after the SPLIT. Source ICF catalogs can also be cloned and read from the target volume. See the COPY command for more information.

6. Start or resume the source Db2 subsystem.

Note: This step is not necessary if you are cloning using consistent split or break mirror.

Access, including modification to source volumes, may be resumed after the COPY step completes successfully. However, if time is not of the essence regarding source volume access, or if the RENAME step runs in a relatively short time frame, it is recommended that source volume access resume only after the RENAME step has completed.

Errors such as volumes not specified that should have been, or catalogs incorrectly specified, are not detected until the RENAME step. These categories of errors require the COPY step to be rerun. If the source volumes have changed since the first running of the COPY step, the same P.I.T. (point-in-time) images cannot be captured.

A typical implementation of the COPY step is to use it as a trigger for an application requiring access to source volumes. This can be accomplished by positioning the COPY job as a predecessor to the application, via a job scheduler.

- If an offline clone was done, start the source Db2 subsystem up. See [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125.
- If an online clone was done, resume the source Db2 subsystem. See [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125.

7. **RENAME step.**

The RENAME step renames and catalogs target volume data sets. The SAFE option, a keyword of the RENAME command is recommended so the RENAME step can be rerun to correct problems caused by mistakes such as incorrectly specified rename masks.

Because of Db2 naming standard requirements, only the Db2 logs and BSDSs can be renamed beyond the hlq. All other Db2 files expect the following:

- Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB01.**
- Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB06.**
- Db2 databases – Db2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn

8. **Db2 cloning procedures.**

Perform one of the cloning procedures found in [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125 before beginning Step 9.

9. **BCSCLEAN step (optional).**

Assuming the Db2 subsystem clone is repetitive, each cycle of the process will likely leave orphaned ICF catalog entries for data sets not used in subsequent cycles of the application.

If the target ICF catalog(s) is dedicated to just the target volume data sets, this problem can be avoided by simply placing an IDCAMS step before the COPY step to delete and redefine the target ICF catalog(s) or use BCSCLEAN to delete catalog entries for data sets Db2 Cloning Tool knows were created during the last cycle. If the redefined ICF catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the ICF catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the catalog.

If the target ICF catalog is used for data sets other than those on target volumes, the BCSCLEAN function will delete catalog entries for data sets Db2 Cloning Tool knows were created during the last cycle. This step can be placed anywhere between the end of target volume access and the next start of the COPY process.

Db2 Cloning Tool process for EMC TimeFinder/Mirror target volume clones

The following table summarizes the steps to accomplish the EMC TimeFinder/Mirror ESTABLISH, SPLIT, and RE-ESTABLISH. More detail is provided in successive topics. For more information, see the EMC TimeFinder Product Guide.

| Step | Description |
|------|---|
| 1 | FINDUCATS step. Optional step to run one time or occasionally, to find user catalogs pointing to source volume data sets. |
| 2 | EMC ESTABLISH step. This step establishes the BCV mirrors. EMC RE-ESTABLISH step. Once the mirror relationship has been established and split, the re-establish will synchronize a previously split pair. |
| 3 | Shut down or suspend the source Db2 subsystem. See Chapter 9, "Cloning Db2 subsystems," on page 125 for additional instructions. These instructions are a prerequisite to Step 4. |
| 4 | EMC SPLIT step. This step splits the mirrors or BCVs. |
| 5 | COPY step. This step backs up the source ICF catalogs that point to the source volume data sets in synchronization with the split. Source ICF catalogs can also be cloned and read off the target volumes. See the COPY command for more information. |
| 6 | Start up or resume the source Db2 subsystem. See Chapter 9, "Cloning Db2 subsystems," on page 125 for additional instructions. These instructions are a prerequisite to Step 7. |
| 7 | RENAME step. This step renames and catalogs target volume data sets. |
| 8 | Db2 Cloning procedure. See Chapter 9, "Cloning Db2 subsystems," on page 125 for additional instructions. |
| 9 | BCSCLEAN step. Optional step runs after target volume usage is terminated and before the next COPY. |

Steps for EMC TimeFinder/Mirror target volume clones

These are the steps to accomplish the EMC TimeFinder/Mirror ESTABLISH, SPLIT, and RE-ESTABLISH.

Procedure

1. FINDUCATS step (optional).

FINDUCATS identifies which user ICF catalogs contain entries for data sets on the source volumes to be copied. The COPY step requires pairs of source/target user ICF catalogs to be specified. FINDUCATS does not negate this need. It is intended to be run prior to initial setup, and possibly on an occasional basis, to make sure the user catalogs that should be specified for the COPY step have not changed.

2. User-supplied step to establish continual mirrors.

This step establishes the mirror relationship between the source and target volumes. It must be placed in the application stream, sufficiently ahead of the desired 'split time', such that the target volumes will be synchronized with their source volume counterparts. If this process is repetitive for each cycle, the TimeFinder/Mirror RE-ESTABLISH can be used for subsequent synchronization. The WAIT parameter waits until the BCVs and the standards are synchronized before the job completes. For more information, see the EMC TimeFinder Product Guide.

The following is a sample of an EMC TimeFinder/Mirror ESTABLISH.

```
//STEP1 EXEC PGM=EMCTF
//STEPLIB DD DISP=SHR,DSN=h1q.EMC.LINKLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
GLOBAL MAXRC=4, WAIT
ESTABLISH 01,4120-4121,4100-4101
ESTABLISH 01,4130,4110
//*
```

The following is a sample of an EMC TimeFinder/Mirror RE-ESTABLISH.

```
//STEP1 EXEC PGM=EMCTF
//STEPLIB DD DISP=SHR,DSN=h1q.EMC.LINKLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
GLOBAL MAXRC=4, WAIT
RE-ESTABLISH 01,4120-4121
RE-ESTABLISH 01,4130
//*
```

3. Shut down or suspend the source Db2 subsystem.

- If an offline clone is desired (Db2 is shut down), choose one of the offline cloning procedures in Chapter 9, “Cloning Db2 subsystems,” on page 125 and complete that procedure prior to beginning Step 4.
- If an online clone is desired (Db2 is suspended), choose one of the online cloning procedures in Chapter 9, “Cloning Db2 subsystems,” on page 125 and complete that procedure prior to beginning Step 4.

These instructions are a prerequisite to Step 4.

4. TimeFinder step to split the mirror relationships.

This step is required to split the mirror relationships to enable processing against the target volumes. The WAIT parameter can elongate the split because it waits until the background process is complete. This would be required if a TimeFinder RESTORE operation is done after the split, but in the case of a volume relabel and data set renames, the NOWAIT parameter is sufficient.

The following is a sample EMC TimeFinder/Consistency Groups consistent SPLIT.

```
//STEP2 EXEC PGM=EMCTF
//STEPLIB DD DISP=SHR,DSN=SYMMI.EMC.TF510.LINKLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
GLOBAL MAXRC=4, NOWAIT
SPLIT 01,4120-4121,CONS(LOCAL(BYP))
SPLIT 01,4130,CONS(LOCAL(BYP))
//*
```

5. COPY step.

The COPY step in this scenario is issued with the 'DATA-MOVER(PGM(NONE))' parameter. 'DATA-MOVER(PGM(NONE))' implies that target volumes have already been created. COPY in this case only backs up the source ICF catalog information needed to rename and catalog the target volume data sets. Optionally, it relabels the target volumes and varies them online for RENAME processing. The COPY should be done immediately after the SPLIT.

The following is a sample COPY command for EMC TimeFinder/Mirror BCVs.

```
//STEP3 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=h1q.SCKZLOAD,DISP=SHR
//CKZINI DD DSN=h1q.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DSN=PRD.DB2A.JRNL,RECOG=KS,KEYLEN=64,KEYOFF=0,DISP=(,CATLG),
// UNIT=SYSALLDA,LRECL=600,SPACE=(CYL,(10,10))
//CKZIN DD *
COPY DATA-MOVER(PGM(NONE)) -
VOLPAIRSDEVN(DB2A01 DB2B01 4120 -
DB2A02 DB2B02 4121 -
```

```

                DB2A03 DB2B03 4130)          -
USERCATALOGS(SRC.USERCAT1 TGT.USERCAT1)    -
CATWORK-DSN(PRD.DB2A.WRK.*)                -
JOURNAL-DDN(JOURNAL)                        -
//*

```

6. Start or resume the source Db2 subsystem.

Access, including modification to source volumes, may be resumed after the COPY step completes successfully. However, if time is not of the essence regarding source volume access, or if the RENAME step runs in a relatively short time frame, it is recommended that source volume access resume only after the RENAME step has completed.

Errors such as volumes not specified that should have been, or catalogs incorrectly specified, are not detected until the RENAME step. These categories of errors require the COPY step to be rerun. If the source volumes have changed since the first running of the COPY step, the same P.I.T. (point-in-time) images cannot be captured.

A typical implementation of the COPY step is to use it as a trigger for an application requiring access to source volumes. This can be accomplished by positioning the COPY job as a predecessor to the application, via a Job Scheduler.

- If an offline clone was done, start the source Db2 subsystem up. See [Chapter 9, “Cloning Db2 subsystems,” on page 125](#).
- If an online clone was done, resume the source Db2 subsystem. See [Chapter 9, “Cloning Db2 subsystems,” on page 125](#).

7. RENAME step.

The RENAME step renames and catalogs target volume data sets. The SAFE option, a keyword of the RENAME command is recommended so the RENAME step can be rerun to correct problems caused by mistakes such as incorrectly specified rename masks.

Because of Db2 naming standard requirements, only the Db2 logs and BSDSs can be renamed beyond the hlq. All other Db2 files expect the following:

- Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB01.**
- Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB06.**
- Db2 databases – Db2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn

8. Db2 cloning procedures.

Perform one of the cloning procedures found in [Chapter 9, “Cloning Db2 subsystems,” on page 125](#) before beginning Step 9.

9. BCSCLEAN step (optional).

Assuming the Db2 subsystem clone is repetitive, each cycle of the process will likely leave orphaned ICF catalog entries for data sets not used in subsequent cycles of the application.

If the target ICF catalog(s) is dedicated to just the target volume data sets, this problem can be avoided by simply placing an IDCAMS step before the COPY step to delete and redefine the target ICF catalog(s) or use BCSCLEAN to delete catalog entries for data sets Db2 Cloning Tool knows were created during the last cycle. If the redefined ICF catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the ICF catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the catalog.

If the target ICF catalog is used for data sets other than those on target volumes, the BCSCLEAN function will delete catalog entries for data sets Db2 Cloning Tool knows were created during the last cycle. This step can be placed anywhere between the end of target volume access and the next start of the COPY process.

Chapter 9. Cloning Db2 subsystems

Separate Db2 cloning procedures by volume are provided for various situations. Choose the appropriate procedure for your situation.

Db2 offline cloning procedures

An offline cloning occurs when Db2 is shut down to clone the data.

- If an offline clone is desired and a Db2 subsystem is being cloned with the intent of a second Db2 subsystem accessing the renamed data sets, refer to the instructions in the topic [“Db2 offline cloning procedure”](#) on page 126.
- If an offline clone of a data sharing group with removal of members is desired, refer to the instructions in the topic [“Db2 offline cloning with removal of data sharing members procedure”](#) on page 128.
- If an offline clone of a data sharing group with the target being non-data sharing is desired, refer to the instructions in the topic [“Db2 offline cloning with target becoming non-data sharing procedure”](#) on page 131.

Db2 online cloning procedures

An online cloning occurs when Db2 SET LOG SUSPEND is issued to suspend transactions, or when a consistent copy process is used such as consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

- If an online clone is desired, refer to the instructions in the topic [“Db2 online cloning procedure”](#) on page 135.
- If an online clone of a data sharing group with removal of members is desired, refer to the instructions in the topic [“Db2 online cloning with removal of data sharing members procedure”](#) on page 140.
- If an online clone of a data sharing group with the target being non-data sharing is desired, refer to the instructions in the topic [“Db2 online cloning with target becoming non-data sharing procedure”](#) on page 146.

Db2 offline or online cloning when the target ICF catalogs reside on target volumes

A Db2 subsystem can be cloned to a target subsystem environment when target ICF catalogs are located on the target volumes. For the Db2 cloning scenarios that are described in the topics that follow, consider whether the target ICF catalogs are located on the target volumes. You can address target ICF catalogs that are located on the target volumes in one of the following ways:

- If target ICF catalogs reside on the target volumes and there are no source volumes that have source ICF catalogs on them, the target ICF catalogs should be manually moved from the target volumes before the volume copy process takes place. The target ICF catalogs can be moved back to the target volumes after the RENAME is complete.
- If target ICF catalogs reside on the target volumes and source ICF catalogs reside on the source volumes, the parameter TARGET-UCATS-ON-TARGET-VOLUMES(Y) can be specified in the cloning job to handle this situation without manual intervention. The target ICF catalogs are allocated to the catalog address space (CAS), potentially on all of the systems in the sysplex. When TARGET-UCATS-ON-TARGET-VOLUMES(Y) is specified, Db2 Cloning Tool instructs the CAS on each system in the sysplex to unallocate the target ICF catalogs prior to the volume copies and the RENAME. The RENAME internally generates the rename masks to rename the source ICF catalog name to the target ICF catalog name for the target volumes. Self-defining information about the copied ICF catalog is updated and all other entries in it are deleted. Thus, the target catalog is overlaid with the source catalog. In addition, the RENAME renames and catalogs all of the target volume data sets to new names.

Db2 offline cloning

An offline Db2 subsystem clone is created by stopping the source Db2 subsystem to achieve your point-in-time copy. Stopping the source Db2 subsystem ensures that all buffers have been flushed, all data has been committed to disk, and that no transactions are in flight.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

If the Db2 system is data sharing and the DDF LOCATION is being changed, when the target Db2 systems are started they might issue the message:

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH THE  
LOCATION NAME location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way Db2 rebuilds the SCA when the target Db2 system is initially started.

Db2 offline cloning procedure

Use this procedure if an offline Db2 subsystem is being cloned (when the source Db2 subsystem has been stopped to achieve your point-in-time copy) with the intent of a second or target Db2 subsystem accessing the renamed data sets.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

A target Db2 subsystem will use the same buffer pool specifications as its corresponding source Db2 subsystem. If the buffer pool definitions in the source Db2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target Db2 subsystem until ALTER BUFFERPOOL commands can be issued.

The names of the work databases are not changed as part of the cloning. The target Db2 system will use the same database names as the source Db2 system. For data sharing, if you want the work database names in the target Db2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

| Step | Description |
|------|---|
| 1 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 2 | Run the COPY command, SCKZJCL member (CKZCOPY). |
| 3 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 4 | Run the RENAME command, SCKZJCL member (CKZREN). |
| 5 | Run the DB2UPDATE command, SCKZJCL member (CKZDUPD). |
| 6 | If Db2 data sharing is used, run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2). |
| 7 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 8 | Run the DB2SQL command, SCKZJCL member (CKZDSQL). |
| 9 | (Optional) Run the DB2SCHEMA-UPDATE command, SCKZJCL member (CKZDSUPD). |

Table 35. Db2 offline cloning procedure (continued)

| Step | Description |
|------|---|
| 10 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 11 | (Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL). |
| 12 | Run the DB2START command, SCKZJCL member (CKZDSTA). |

Steps for Db2 offline cloning

These are the steps for Db2 offline cloning.

Procedure

1. Run the **DB2STOP** command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the source Db2 subsystem. This ensures that buffers have been flushed, all data has been committed to disk, and no transactions are in flight.

2. Run the **COPY** command.

Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend', then run the COPY command using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied, and optionally, relabel and vary online the target volumes.

3. Run the **DB2START** command.

Run the DB2START command using SCKZJCL member (CKZDSTA) to start the source Db2 subsystem. At this point, the data has been cloned and the source volumes are no longer required.

4. Run the **RENAME** command.

Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. Run the **DB2UPDATE** command.

Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE updates the Db2 directory and the Db2 boot strap data sets (BSDSs).

- Db2 directory updates – the VCATNAME, and optionally, the Db2 storage group names are updated.
- BSDS updates – the Db2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

If data sharing is used in your environment, the target Db2 XCF structures need to be deallocated. The DB2®-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem.

6. If Db2 data sharing is used, run **DB2UPDATE** again.

If Db2 data sharing is used in your environment, run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE

BSDONLY command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any Db2 directory updates.

7. Run the DB2START command.

To start the target Db2 subsystem in maintenance mode using the special zparms, run the DB2START command. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

8. Run the DB2SQL command to update the Db2 catalog.

This step assumes that the plan and package have already been installed. Run the DB2SQL command on the target subsystem using SCKZJCL member (CKZDSQL). The DB2SQL command makes the necessary changes to the Db2 catalog. The changes include the VCATNAME, storage group names, volumes, and the GROUP_MEMBER field value in SYSIBM.SYSDATABASE for work databases, if data sharing is used.

9. (Optional) Run the DB2SCHEMA-UPDATE command.

This step assumes that the plan and package have already been installed. To update and re-create objects using new schema values, run the DB2SCHEMA-UPDATE command on the target subsystem using SCKZJCL member (CKZDSUPD). The DB2SCHEMA-UPDATE command changes the creator, owner, and the schema of database objects.

10. Run the DB2STOP command.

Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO) to stop the target Db2 subsystem from running in maintenance mode.

11. (Optional) Run the DB2UTILXCLEAN command.

To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source Db2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source Db2 subsystem was cloned.

12. Run the DB2START command.

Run the DB2START command using SCKZJCL member (CKZDSTA) to start the target Db2 subsystem again, with its normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.

Db2 offline cloning with removal of data sharing members procedure

Db2 offline cloning infers that the source Db2 subsystem has been stopped to achieve your point-in-time copy. It ensures that all buffers have been flushed, all data has been committed to disk, and that no transactions are in flight.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

This procedure requires the following:

- The new target Db2 data sharing group members will have no log history, so image copies must be taken if further recovery is desired.
- For a data sharing environment, the target Db2 XCF structures should be deallocated prior to the first starting of the target Db2 subsystem.
- Only the Db2 members being retained need to be cloned.
- The RENAME-MASKS keyword in the RENAME command needs to include entries that will cause the renaming of all the member BSDSs.
- The names of the work databases are not changed as part of the cloning. The target Db2 system will use the same database names as the source Db2 system. For data sharing, if you want the work database

names in the target Db2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for offline cloning with removal of data sharing members:

| <i>Table 36. Db2 offline cloning with removal of data sharing members procedure</i> | |
|---|--|
| Step | Description |
| 1 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 2 | Run the COPY command, SCKZJCL member (CKZCOPY). |
| 3 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 4 | Run the RENAME command, SCKZJCL member (CKZREN). |
| 5 | Run the DB2UPDATE command, SCKZJCL member (CKZDUPD). |
| 6 | Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2). |
| 7 | Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS). |
| 8 | Run the DB2LGRNXCLEAN command, SCKZJCL member (CKZDLGCL). |
| 9 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 10 | Run the DB2SQL command, SCKZJCL member (CKZDSQL). |
| 11 | (Optional) Run the DB2SCHEMA-UPDATE command, SCKZJCL member (CKZDSUPD). |
| 12 | Set the Db2 buffer pool attributes. |
| 13 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 14 | (Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL). |
| 15 | Run the DB2START command, SCKZJCL member (CKZDSTA), to start the target Db2 subsystem again, using its normal zparms DSNZPARx. |
| 16 | Run the DB2START command, SCKZJCL member (CKZDSTA), to start the other target Db2 subsystems, using their normal zparms DSNZPARx. |
| 17 | Set the Db2 buffer pool attributes. |

Steps for Db2 offline cloning with removal of data sharing members

These are the steps for Db2 offline cloning with removal of data sharing members.

Procedure

1. Run the **DB2STOP** command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the source Db2 subsystem. This ensures that buffers have been flushed, all data has been committed to disk, and no transactions are in flight.

2. Run the **COPY** command.

Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using IBM FlashCopy or SnapShot, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,” on page 371 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split

or suspend', then run the COPY command using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied, and optionally, relabel and vary online the target volumes.

3. Run the DB2START command.

Run the DB2START command on the source system using SCKZJCL member (CKZDSTA). At this point, the data has been cloned and the source volumes are no longer required.

4. Run the RENAME command.

Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. Run the DB2UPDATE command.

Run DB2UPDATE command using SCKZJCL member (CKZDUPD). The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE updates the Db2 directory and the Db2 boot strap data sets (BSDSs).

- Db2 directory updates – the VCATNAME, and optionally, the Db2 storage group names are updated.
- BSDS updates – the Db2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target Db2 XCF structures and group members need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem.

6. Run the DB2UPDATE command again.

Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member that is being cloned. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE BSDSONLY command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any Db2 directory updates.

The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

7. Run the DB2RBLDBSDS command.

Run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for each data sharing member that is being cloned. This step will rebuild the member BSDS to have only active logs in it.

8. Run the DB2LGRNXCLEAN command.

Run the DB2LGRNXCLEAN command using SCKZJCL member (CKZDLGCL). This step will clean out all information in SYSLGRNX.

9. Run the DB2START command.

Run the DB2START command to start the primary target in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

10. Run the DB2SQL command.

This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic [“Cloning a Db2 subsystem”](#) on page 112.

Run the DB2SQL command on the target subsystem using SCKZJCL member (CKZDSQL). The DB2SQL command makes the necessary changes to the Db2 catalog. The changes include the VCATNAME, storage group names, volumes, and the GROUP_MEMBER field value in SYSIBM.SYSDATABASE for work databases.

11. (Optional) Run the DB2SCHEMA-UPDATE command.

This step assumes that the plan and package have already been installed. To update and re-create objects using new schema values, run the DB2SCHEMA-UPDATE command on the target subsystem

using SCKZJCL member (CKZDSUPD). The DB2SCHEMA-UPDATE command changes the creator, owner, and the schema of database objects.

12. Set the Db2 buffer pool attributes.

The Db2 buffer pool attributes need to be set to the desired values for the primary target Db2 subsystem. The Db2 -ALTER BUFFERPOOL command can be used to set desired buffer pool attributes.

13. Run the DB2STOP command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the primary target Db2 subsystem from running in maintenance mode.

14. (Optional) Run the DB2UTILXCLEAN command.

To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source Db2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source Db2 subsystem was cloned.

15. Run the DB2START command.

Run the DB2START command using SCKZJCL member (CKZDSTA) to start the primary target Db2 subsystem again, with its normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.

16. Run the DB2START command.

Run the DB2START command using SCKZJCL member (CKZDSTA) to start the other target Db2 subsystems that are being cloned, with their normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.

17. Set the Db2 buffer pool attributes.

The Db2 buffer pool attributes need to be set to the desired values for the other target Db2 subsystems that are being cloned. The Db2 -ALTER BUFFERPOOL command can be used to set desired buffer pool attributes.

Db2 offline cloning with target becoming non-data sharing procedure

Db2 offline cloning infers that the source Db2 subsystem has been stopped to achieve your point-in-time copy. It ensures that all buffers have been flushed, all data has been committed to disk, and that no transactions are in flight.

Important: For Db2 11, there are several conditions that must be met prior to the target Db2 becoming non-data sharing. If these conditions are not met, the target Db2 will not be usable.

- The source data sharing group must be in new-function mode.
- The BSDS of the surviving target subsystem must be enabled for extended RBA and LRSN.
- The Db2 directory and catalog must be converted to 10-byte extended format.
- All other objects (table and index spaces) must be converted to 10-byte extended format.

Additional information can be found in the IBM Knowledge Center topic "Disabling data sharing", in the Db2 11 for z/OS online documentation for installing and migrating Db2.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

This procedure requires the following:

- As the target Db2 subsystem is to become non-data sharing, both its special and normal zparms need to be non-data sharing.

- For a data sharing environment, the target Db2 XCF structures should be deallocated prior to the first starting of the target Db2 subsystem.
- Only the Db2 member being retained needs to be cloned.
- The RENAME-MASKS keyword in the RENAME command needs to include entries that will cause the renaming of all the member BSDSs.
- A target Db2 subsystem will use the same buffer pool specifications as its corresponding source Db2 subsystem. If the buffer pool definitions in the source Db2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target Db2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target Db2 system will use the same database names as the source Db2 system. For data sharing, if you want the work database names in the target Db2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for offline cloning with the target becoming a non-data sharing subsystem:

| Step | Description |
|------|---|
| 1 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 2 | Run the COPY command, SCKZJCL member (CKZCOPY). |
| 3 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 4 | Run the RENAME command, SCKZJCL member (CKZREN). |
| 5 | Run the DB2UPDATE command, SCKZJCL member (CKZDUPD). |
| 6 | Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2). |
| 7 | Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS). |
| 8 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 9 | Run the DB2SQL command, SCKZJCL member (CKZDSQL), to update the Db2 catalog. |
| 10 | (Optional) Run the DB2SCHEMA-UPDATE command, SCKZJCL member (CKZDSUPD). |
| 11 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 12 | (Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL). |
| 13 | Run the DB2START command, SCKZJCL member (CKZDSTA). |

Steps for Db2 offline cloning with target becoming non-data sharing

These are the steps for Db2 offline cloning with the target becoming non-data sharing.

Procedure

1. Run the **DB2STOP** command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the source Db2 subsystem. This ensures that buffers have been flushed, all data has been committed to disk, and no transactions are in flight.

2. Run the **COPY** command.

Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend', then run the COPY command using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied, and optionally, relabel and vary online the target volumes.

3. Run the DB2START command.

Run the DB2START command on the source system using SCKZJCL member (CKZDSTA). At this point, the data has been cloned and the source volumes are no longer required.

4. Run the RENAME command.

Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets. The RENAME-MASKS keyword needs to include entries that will cause the renaming of all the member BSDSs.

5. Run the DB2UPDATE command.

Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used. The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE updates the Db2 directory and the Db2 boot strap data sets (BSDSs).

- Db2 directory updates – the VCATNAME, and optionally, the Db2 storage group names are updated.
- BSDS updates – the Db2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target Db2 XCF structures and group members need to be deallocated. The DB2-XFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem.

6. Run the DB2UPDATE command again.

Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE BSDSONLY command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any Db2 directory updates.

The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

7. Run the DB2RBLDBSDS command.

Run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for the desired target member using the keyword DATA-SHARING(NO). This step will alter the member BSDS to be non-data sharing.

8. Run the DB2START command.

Run the DB2START command to start the primary target Db2 subsystem in maintenance mode using the special non-data sharing zparms. Use the SPECIAL, DSNZPARM, and REPLY-TO-RESTART-WTOR(Y) keywords on the target subsystem using SCKZJCL member (CKZDSTA) to start the target Db2 subsystem.

This start-up of the target Db2 subsystem will require a cold start, so an operator reply will need to be made to the Db2 message:

```
DSNJ246I  CONDITIONAL RESTART RECORD INDICATES COLD START AT
RBA xxxxxxxxxxxx.  REPLY Y TO CONTINUE, N TO CANCEL
```

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command will cause DB2START to automatically reply to the DSNJ246I WTOR message.

9. Run the DB2SQL command.

This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic [“Cloning a Db2 subsystem”](#) on page 112.

Run the DB2SQL command on the target subsystem using SCKZJCL member (CKZDSQL). The DB2SQL command makes the necessary changes to the Db2 catalog. The changes include the VCATNAME, storage group names, volumes, and the GROUP_MEMBER field value in SYSIBM.SYSDATABASE for work databases.

10. (Optional) Run the DB2SCHEMA-UPDATE command.

This step assumes that the plan and package have already been installed. To update and re-create objects using new schema values, run the DB2SCHEMA-UPDATE command on the target subsystem using SCKZJCL member (CKZDSUPD). The DB2SCHEMA-UPDATE command changes the creator, owner, and the schema of database objects.

11. Run the DB2STOP command.

Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO) to stop the target Db2 subsystem from running in maintenance mode.

12. (Optional) Run the DB2UTILXCLEAN command.

To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source Db2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source Db2 subsystem was cloned.

13. Run the DB2START command.

Run the DB2START command using SCKZJCL member (CKZDSTA) to start the target Db2 subsystem again, with its normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.

Db2 online cloning

An online Db2 subsystem clone is created by suspending the source Db2 subsystem to achieve your point-in-time copy. By suspending the source Db2 subsystem, any pending database writes are forced to disk, update activity is suspended, and the log buffers are flushed to disk. An alternative to suspending the source Db2 subsystem is to use consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror to achieve your point-in-time copy.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

If the Db2 system is data sharing and the DDF LOCATION is being changed, when the target Db2 systems are started, they might issue the message:

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH  
THELOCATION NAME location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way Db2 rebuilds the SCA when the target Db2 system is initially started.

Important: In order to prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem”](#) on page 112 for information about how to set up the special dsnzparm module.

Status of transactions in flight

An online cloning solution often results in transactions in flight. These in-flight transactions, cloned to the target subsystem, result in the same target subsystem action that would happen on the source system if it were to have died at that same time and then restarted. When you use online cloning, the target restart is essentially an emergency restart of a failed system.

The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

The 'unit of work' or transaction can be in any one of the following states:

In flight

This is where a transaction is most of the time. On the target subsystem it will be backed out to the last commit point. Read-only transactions have nothing to back out.

Commit

The transaction is in the process of taking a commit. On the target subsystem the transaction updates should be committed.

Abort

The transaction is in the process of aborting. On the target subsystem it will be backed out to the last commit point

In doubt

The transaction was committing and was between phase 1 and phase 2 commit processing. Db2 does not know if the transaction should be backed out or committed. Manual intervention is required to either back out or commit the transaction.

Log data needed to back out a transaction should be contained in the active logs. It is possible that a back-out will need log data that no longer resides in an active log. In this case, archive logs will be needed to successfully complete the back-out. Whether and which archive logs will be necessary for restart of the target subsystem depends on how busy the source system is and how large the active logs are. We recommend the point-in-time copy created online (via Db2 SET LOG SUSPEND, consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror) be done at a quiet time.

If there are DDF transactions active during the cloning, when the target Db2 systems are started they may issue the message:

```
DSNL034E DDF CANNOT BE STARTED BECAUSE OF BSDS INCONSISTENCIES
```

To resolve this condition and allow DDF to start, the following Db2 command must be issued on the target Db2 system:

```
-RESET INDOUBT LUNAME(*) FORCE
```

This Db2 command should not be issued until after the DB2SQL command has been run.

Db2 online cloning procedure

If an online clone is desired, refer to these instructions.

An online clone can be done with one of these methods:

- Using Db2 SET LOG SUSPEND and RESUME to suspend and resume transactions during the cloning process.
- Using consistent FlashCopy, SnapShot, or TimeFinder/Clone to manage I/O to the volumes during the cloning process. The FlashCopy, SnapShot, or TimeFinder/Clone API manages the suspension and resuming of activity against the volumes during cloning.
- Using consistent split or break mirror. The split or break mirror API manages the suspension and resuming of activity against the volumes.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use

the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem” on page 112](#) for information on how to set up the special dsnzparm module.

This procedure requires the following:

- If a Db2 data sharing group is being cloned, all members of the group should also be cloned.
- For a data sharing environment, the target Db2 XCF structures should also be deallocated prior to the first starting of the target Db2 subsystem.
- For a Db2 data sharing environment, the special zparks will need to be set up for each target member. For more information, see the topic [“Cloning a Db2 subsystem” on page 112](#).
- Because the source Db2 subsystems are active at the time of the clone, any one of them may have a lock outstanding on a Db2 table. These outstanding locks are carried into the target Db2 sharing environment. To free these locks in the target environment, the corresponding target member will need to be started in maintenance mode with its own special zparks.
- A target Db2 subsystem will use the same buffer pool specifications as its corresponding source Db2 subsystem. If the buffer pool definitions in the source Db2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target Db2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target Db2 system will use the same database names as the source Db2 system. For data sharing, if you want the work database names in the target Db2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for online cloning:

| Step | Description |
|-------------|--|
| 1 | Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror. |
| 2 | Run the COPY command, SCKZJCL member (CKZCOPY). |
| 3 | Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror. |
| 4 | Run the RENAME command, SCKZJCL member (CKZREN). |
| 5 | Run the DB2UPDATE command, SCKZJCL member (CKZDUPD). |
| 6 | If Db2 data sharing is used, run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2). |
| 7 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 8 | If Db2 data sharing is used, run the DB2START command, SCKZJCL member (CKZDSTA), to start the remaining target Db2 members. |
| 9 | Run the DB2FIX command, SCKZJCL member (CKZDFIX). |

Table 38. Db2 online cloning procedure (continued)

| Step | Description |
|------|--|
| |  Attention: Steps 10, 11, and 12 are required only if the table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and must be redone. Otherwise, proceed from step 9 directly to step 13. |
| 10 | (Optional, see note.) Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 11 | (Optional, see note.) Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD). |
| 12 | (Optional, see note.) Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 13 | Run the DB2SQL command, SCKZJCL member (CKZDSQL). |
| 14 | Run the DB2FIX command, SCKZJCL member (CKZDFIX). |
| 15 | (Optional) Run the DB2SCHEMA-UPDATE command, SCKZJCL member (CKZDSUPD). |
| 16 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 17 | (Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL). |
| 18 | Run the DB2START command, SCKZJCL member (CKZDSTA). |

Steps for Db2 online cloning

These are the steps for Db2 online cloning.

Procedure

1. Run the **DB2SETLOG** command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

The Db2 Cloning Tool DB2SETLOG command with the SUSPEND keyword issues a SET LOG LOGLOAD(0) command to force any pending database writes to disk, followed by a SET LOG SUSPEND command to suspend update activity and flush the log buffers to disk. On the source system, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL).

If you are running in a data sharing environment, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

2. Run the **COPY** command.

Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend' then run COPY using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If a Db2 data sharing group is being cloned, all members of the group should also be cloned.

3. Run the **DB2SETLOG** command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

After the data has been copied to the target volumes, the source ICF catalogs have been backed up, and COPY has completed successfully, you can resume access to the source volumes.

Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), on the source system to resume update activity. If you are running in a data sharing environment, run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

4. Run the RENAME command.

Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. Run the DB2UPDATE command.

Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE updates the Db2 directory and the Db2 boot strap data sets (BSDSs).

- Db2 directory updates – the VCATNAME, and optionally, the Db2 storage group names are updated.
- BSDS updates – the Db2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

If Db2 data sharing is used in your environment, the target Db2 XCF structures need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem. See [“Deallocating target Db2 coupling facility structures”](#) on page 1160 for more information on this process.

6. If Db2 data sharing is used, run the DB2UPDATE command again.

If Db2 data sharing is used in your environment, run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any Db2 directory updates.

7. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem”](#) on page 112 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target subsystem in maintenance mode using the special zparks. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

Note: The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

The following error messages are expected. If an in-flight transaction was in progress, multiple messages may be associated with it:

```
DSNI001I @TGT1 RESTART HAS BEEN DEFERRED
          REASON 00C90095
          TYPE 00000200
          NAME databasename .tablespace or indexspace name
DSNB250E @TGT1 DSNIIIMPD A PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST
          DATABASE NAME=database name
          SPACE NAME=tablespace or indexspace name
          DATA SET NUMBER=1
          PAGE RANGE X'nnnnnnnn' TO X'nnnnnnnn'
          START LRSN=X'nnnnnnnnnnnn'
          END LRSN=X'nnnnnnnnnnnn'
          START RBA=X'nnnnnnnnnnnn'
```

8. If Db2 data sharing is used, run the DB2START command again.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem” on page 112](#) for information on how to set up the special dsnzparm module.

If Db2 data sharing is used in your environment, the other target Db2 members may hold locks that need to be released before the DB2FIX command can be run. To release these locks, start the remaining target Db2 members in maintenance mode with the special zparks and then stop the members, as follows:

- Run the DB2START command with the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).
- Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO).

The DB2START and DB2STOP commands can be run in the same job.

Note: The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

9. Run the DB2FIX command.

Run the DB2FIX command to correct any of the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) page spaces that are restricted.

Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(Db2) on the target subsystem. This will start any page spaces in the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) that have LPL or GRECP status.

If Db2 data sharing is used in your environment, only one Db2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.



Attention: Steps 10, 11, and 12 need to be performed ONLY if the table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and need to be redone. These steps can be automated by using either the DB2FIX return code setting, or the DB2FIX WTO message produced when DB2FIX starts restricted page spaces. For more information, see the [“DB2FIX” on page 420](#) command. Otherwise, proceed to step 13.

10. Run the DB2STOP command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the target Db2 subsystem from running in maintenance mode.

11. Run the DB2UPDATE command again.

Run the DB2UPDATE command with the DBD01ONLY keyword using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE with the DBD01ONLY keyword updates only the Db2 directory. The VCATNAME, and optionally, the Db2 storage group names are updated.

If Db2 data sharing is used in your environment, the target Db2 XCF structures need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem. See [“Deallocating target Db2 coupling facility structures” on page 1160](#) for more information on this process.

12. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem” on page 112](#) for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target Db2 subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARAM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

13. Run the DB2SQL command.

Run the DB2SQL command using SCKZJCL member (CKZDSQL) to update the Db2 catalog. This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic [“Cloning a Db2 subsystem” on page 112](#).

The DB2SQL command makes the necessary changes to the Db2 catalog. The changes include the VCATNAME, storage group names, volumes, and the GROUP_MEMBER field value in SYSIBM.SYSDATABASE for work databases, if data sharing is used.

14. Run the DB2FIX command.

Run the DB2FIX command to correct any of the application page spaces that are restricted. Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(APPLICATION) on the target subsystem. This will start any application page spaces that have LPL or GRECP status.

If Db2 data sharing is used in your environment, only one Db2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

15. (Optional) Run the DB2SCHEMA-UPDATE command.

This step assumes that the plan and package have already been installed. To update and re-create objects using new schema values, run the DB2SCHEMA-UPDATE command on the target subsystem using SCKZJCL member (CKZDSUPD). The DB2SCHEMA-UPDATE command changes the creator, owner, and the schema of database objects.

16. Run the DB2STOP command.

To stop the target Db2 subsystem from running in maintenance mode, run the DB2STOP command using SCKZJCL member (CKZDSTO).

17. (Optional) Run the DB2UTILXCLEAN command.

To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source Db2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source Db2 subsystem was cloned.

18. Run the DB2START command.

To start the target Db2 subsystem using its 'normal' zparms, run the DB2START command using SCKZJCL member (CKZDSTA). You can also start other members of the target data sharing group with their normal zparms by running DB2START on those members.

Db2 online cloning with removal of data sharing members procedure

Refer to these instructions if an online clone of a data sharing group with removal of members is desired. An online cloning occurs when Db2 SET LOG SUSPEND is issued to suspend transactions, or when a consistent copy process is used such as consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use

the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem” on page 112](#) for information on how to set up the special dsnzparm module.

This procedure requires the following:

- All members of the group should also be cloned.
- The target Db2 XCF structures should be deallocated prior to the first starting of the target Db2 subsystem.
- The new target Db2 data sharing group members will have no log history, so image copies must be taken if further recovery is desired.
- The special zparms will need to be set up for each target member. For more information, see the topic [“Cloning a Db2 subsystem” on page 112](#).
- Because the source Db2 subsystems are active at the time of the clone, any one of them may have a lock outstanding on a Db2 table. These outstanding locks are carried into the target Db2 sharing environment. To free these locks in the target environment, the corresponding target member will need to be started in maintenance mode with its own special zparms.
- A target Db2 subsystem will use the same buffer pool specifications as its corresponding source Db2 subsystem. If the buffer pool definitions in the source Db2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target Db2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target Db2 system will use the same database names as the source Db2 system. For data sharing, if you want the work database names in the target Db2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for online cloning with removal of data sharing members:

| <i>Table 39. Db2 online cloning with removal of data sharing members procedure</i> | |
|--|--|
| Step | Description |
| 1 | Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror. |
| 2 | Run the COPY command, SCKZJCL member (CKZCOPY). |
| 3 | Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror. |
| 4 | Run the RENAME command, SCKZJCL member (CKZREN). |
| 5 | Run the DB2UPDATE command, SCKZJCL member (CKZDUPD). |
| 6 | Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2). |
| 7 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 8 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 9 | Run the DB2FIX command, SCKZJCL member (CKZDFIX), to correct any of the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) page spaces that are restricted. |

Table 39. Db2 online cloning with removal of data sharing members procedure (continued)

| Step | Description |
|------|--|
| |  Attention: Steps 10, 11, and 12 are required only if the table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSND01 was restricted when DB2FIX ran. The changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and must be redone. Otherwise, proceed from step 10 directly to step 13. |
| 10 | (Optional, see note.) Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 11 | (Optional, see note.) Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD). |
| 12 | (Optional, see note.) Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 13 | Run the DB2SQL command, SCKZJCL member (CKZDSQL). |
| 14 | Run the DB2FIX command, SCKZJCL member (CKZDFIX), to correct any of the application page spaces that are restricted.. |
| 15 | (Optional) Run the DB2SCHEMA-UPDATE command, SCKZJCL member (CKZDSUPD). |
| 16 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 17 | Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS). |
| 18 | Run the DB2LGRNXCLEAN command, SCKZJCL member (CKZDLGCL). |
| 19 | Run the DB2XCFCLEAN command, SCKZJCL member (CKZDXCFC). |
| 20 | (Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL). |
| 21 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 22 | Set the Db2 buffer pool attributes. |

Steps for Db2 online cloning with removal of data sharing members

These are the steps for Db2 online cloning with removal of data sharing members.

Procedure

1. Run the **DB2SETLOG** command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

The Db2 Cloning Tool DB2SETLOG command with the SUSPEND keyword issues a SET LOG LOGLOAD(0) command to force any pending database writes to disk, followed by a SET LOG SUSPEND command to suspend update activity and flush the log buffers to disk. On the source system, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL).

Run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

2. Run the **COPY** command.

Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,” on page 371 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend' then run COPY using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied.

All members of the Db2 data sharing group should be cloned.

3. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

After the data has been copied to the target volumes, the source ICF catalogs have been backed up, and COPY has completed successfully, you can resume access to the source volumes.

Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), on the source system to resume update activity. Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

4. Run the RENAME command.

Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. Run the DB2UPDATE command.

Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE updates the Db2 directory and the Db2 boot strap data sets (BSDSs).

- Db2 directory updates – the VCATNAME, and optionally, the Db2 storage group names are updated.
- BSDS updates – the Db2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target Db2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem. See [“Deallocating target Db2 coupling facility structures”](#) on page 1160 for more information on this process.

6. Run the DB2UPDATE command again.

Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any Db2 directory updates.

7. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL.. Refer to the topic [“Cloning a Db2 subsystem”](#) on page 112 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target subsystem in maintenance mode using the special zparks. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

Note: The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

The following error messages are expected. If an in-flight transaction was in progress, multiple messages may be associated with it:

```
DSNI001I  @TGT1 RESTART HAS BEEN DEFERRED
          REASON 00C90095
          TYPE 00000200
          NAME databasename .tablespace or indexspace name
```

```

DSNB250E @TGT1 DSNIIIMPD A PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST
        DATABASE NAME=database name
        SPACE NAME=tablespace or indexspace name
        DATA SET NUMBER=1
        PAGE RANGE X'nnnnnnnn' TO X'nnnnnnnn'
        START LRSN=X'nnnnnnnnnnnn'
        END LRSN=X'nnnnnnnnnnnn'
        START RBA=X'nnnnnnnnnnnn'

```

8. Run the DB2START command again.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL.. Refer to the topic [“Cloning a Db2 subsystem” on page 112](#) for information on how to set up the special dsnzparm module.

The other target Db2 members may hold locks that need to be released before the DB2FIX command can be run. To release these locks, start the remaining target Db2 members in maintenance mode with the special zparms and then stop the members, as follows:

- Run the DB2START command with the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).
- Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO).

The DB2START and DB2STOP commands can be run in the same job.

Note: The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

9. Run the DB2FIX command.

Run the DB2FIX command to correct any of the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) page spaces that are restricted.

Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(Db2) on the target subsystem. This will start any page spaces in the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) that have LPL or GRECP status.

Only one Db2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.



Attention: Steps 10, 11, and 12 need to be performed ONLY if the table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and need to be redone. These steps can be automated by using either the DB2FIX return code setting, or the DB2FIX WTO message produced when DB2FIX starts restricted page spaces. For more information, see the [“DB2FIX” on page 420](#) command. Otherwise, proceed to step 13.

10. Run the DB2STOP command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the target Db2 subsystem from running in maintenance mode.

11. Run the DB2UPDATE command again.

Run the DB2UPDATE command with the DBD01ONLY keyword using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE with the DBD01ONLY keyword updates only the Db2 directory. The VCATNAME, and optionally, the Db2 storage group names are updated.

The target Db2 XCF structures need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2

subsystem. See [“Deallocating target Db2 coupling facility structures” on page 1160](#) for more information on this process.

12. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL.. Refer to the topic [“Cloning a Db2 subsystem” on page 112](#) for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target Db2 subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

13. Run the DB2SQL command.

Run the DB2SQL command using SCKZJCL member (CKZDSQL) to update the Db2 catalog. This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic [“Cloning a Db2 subsystem” on page 112](#).

The DB2SQL command makes the necessary changes to the Db2 catalog. The changes include the VCATNAME, storage group names, volumes, and the GROUP_MEMBER field value in SYSIBM.SYSDATABASE for work databases.

14. Run the DB2FIX command.

Run the DB2FIX command to correct any of the application page spaces that are restricted. Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(APPLICATION) on the target subsystem. This will start any application page spaces that have LPL or GRECP status.

Only one Db2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

15. (Optional) Run the DB2SCHEMA-UPDATE command.

This step assumes that the plan and package have already been installed. To update and re-create objects using new schema values, run the DB2SCHEMA-UPDATE command on the target subsystem using SCKZJCL member (CKZDSUPD). The DB2SCHEMA-UPDATE command changes the creator, owner, and the schema of database objects.

16. Run the DB2STOP command.

To stop the target Db2 subsystem from running in maintenance mode, run the DB2STOP command using SCKZJCL member (CKZDSTO).

17. Run the DB2RBLDBSDS command.

Run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for each desired target data sharing member. This step will rebuild the member BSDS to have only active logs in it.

18. Run the DB2LGRNXCLEAN command.

To clean out all information in SYSLGRNX, run the DB2LGRNXCLEAN command using SCKZJCL member (CKZDLGCL).

19. Run the DB2XCFCLEAN command.

Run the DB2XCFCLEAN command using SCKZJCL member (CKZDXCFC). This step will deallocate the Db2 XCF structures and remove the XCF group members.

20. (Optional) Run the DB2UTILXCLEAN command.

To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source Db2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source Db2 subsystem was cloned.

21. Run the DB2START command.

To start the target Db2 subsystem using its 'normal' zparms, run the DB2START command using SCKZJCL member (CKZDSTA). You can also start other members of the target data sharing group with their normal zparms by running DB2START on those members.

22. Set the Db2 buffer pool attributes.

The Db2 buffer pool attributes need to be set to the desired values. The Db2 -ALTER BUFFERPOOL command can be used to set desired buffer pool attributes.

Db2 online cloning with target becoming non-data sharing procedure

Refer to these instructions if an online clone of a data sharing group with the target becoming non-data sharing is desired. An online cloning occurs when Db2 SET LOG SUSPEND is issued to suspend transactions, or when a consistent copy process is used such as consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

Important: For Db2 11, there are several conditions that must be met prior to the target Db2 becoming non-data sharing. If these conditions are not met, the target Db2 will not be usable.

- The source data sharing group must be in new-function mode.
- The BSDS of the surviving target subsystem must be enabled for extended RBA and LRSN.
- The Db2 directory and catalog must be converted to 10-byte extended format.
- All other objects (table and index spaces) must be converted to 10-byte extended format.

Additional information can be found in the IBM Knowledge Center topic "Disabling data sharing", in the Db2 11 for z/OS online documentation for installing and migrating Db2.

Important: If utilities might be running or registered in SYSUTILX on the source Db2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when Db2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source Db2 subsystem might become corrupted when the target Db2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic ["Cloning a Db2 subsystem" on page 112](#) for information on how to set up the special dsnzparm module.

This procedure requires the following:

- All members of the data sharing group should be cloned.
- The target Db2 XCF structures should be deallocated prior to the first starting of the target Db2 subsystem.
- The initial start-up of the target Db2 system must be done in data sharing mode. This is necessary because this is an online cloning and the resolution of work in flight by the target Db2 must be done in the same mode as the source Db2.
- The special zparms will need to be set up for each target member. For more information, see the topic ["Cloning a Db2 subsystem" on page 112](#).
- Because the source Db2 subsystems are active at the time of the clone, any one of them may have a lock outstanding on a Db2 table. These outstanding locks are carried into the target Db2 sharing environment. To free these locks in the target environment, the corresponding target member will need to be started in maintenance mode with its own special zparms.
- The special zparms used by the Db2 subsystems need to be data sharing and the normal zparms used by the final target Db2 subsystems needs to be non-data sharing.
- A target Db2 subsystem will use the same buffer pool specifications as its corresponding source Db2 subsystem. If the buffer pool definitions in the source Db2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target Db2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target Db2 system will use the same database names as the source Db2 system. For data sharing, if you want the work database

names in the target Db2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for online cloning with the target becoming a non-data sharing subsystem:

| <i>Table 40. Db2 online cloning with target becoming non-data sharing procedure</i> | |
|---|--|
| Step | Description |
| 1 | Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror. |
| 2 | Run the COPY command, SCKZJCL member (CKZCOPY). |
| 3 | Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror. |
| 4 | Run the RENAME command, SCKZJCL member (CKZREN). |
| 5 | Run the DB2UPDATE command, SCKZJCL member (CKZDUPD). |
| 6 | Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2). |
| 7 | Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 8 | Run the DB2START command, SCKZJCL member (CKZDSTA), to start the remaining target Db2 members. |
| 9 | Run the DB2FIX command, SCKZJCL member (CKZDFIX). |
| |  Attention: Steps 10, 11, and 12 are required only if the table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and must be redone. Otherwise, proceed from step 9 directly to step 13. |
| 10 | <i>(Optional, see note.)</i> Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 11 | <i>(Optional, see note.)</i> Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD). |
| 12 | <i>(Optional, see note.)</i> Run the DB2START command, SCKZJCL member (CKZDSTA). |
| 13 | Run the DB2SQL command, SCKZJCL member (CKZDSQL). |
| 14 | Run the DB2FIX command, SCKZJCL member (CKZDFIX), to correct any of the application page spaces that are restricted. |
| 15 | <i>(Optional)</i> Run the DB2SCHEMA-UPDATE command, SCKZJCL member (CKZDSUPD). |
| 16 | Run the DB2STOP command, SCKZJCL member (CKZDSTO). |
| 17 | Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS). |
| 18 | Run the DB2XCFCLEAN command, SCKZJCL member (CKZDXCFC). |
| 19 | <i>(Optional)</i> Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL). |
| 20 | Run the DB2START command, SCKZJCL member (CKZDSTA). |

Steps for Db2 online cloning with target becoming non-data sharing

These are the steps for Db2 online cloning with target becoming non-data sharing.

Procedure

1. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

The Db2 Cloning Tool DB2SETLOG command with the SUSPEND keyword issues a SET LOG LOGLOAD(0) command to force any pending database writes to disk, followed by a SET LOG SUSPEND command to suspend update activity and flush the log buffers to disk. On the source system, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL).

Run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

2. Run the COPY command.

Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend' then run COPY using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied.

All members of the Db2 data sharing group should be cloned.

3. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

After the data has been copied to the target volumes, the source ICF catalogs have been backed up, and COPY has completed successfully, you can resume access to the source volumes.

Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), on the source system to resume update activity. Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

4. Run the RENAME command.

Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. Run the DB2UPDATE command.

Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE updates the Db2 directory and the Db2 boot strap data sets (BSDSs).

- Db2 directory updates – the VCATNAME, and optionally, the Db2 storage group names are updated.
- BSDS updates – the Db2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target Db2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2

subsystem. See [“Deallocating target Db2 coupling facility structures”](#) on page 1160 for more information on this process.

6. Run the DB2UPDATE command again.

Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any Db2 directory updates.

The source Db2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

7. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to [“Cloning a Db2 subsystem”](#) on page 112 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target subsystem in maintenance mode using the special zparks. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

This start-up of the target Db2 subsystem must be done in data sharing mode. This is necessary because this is an online cloning and the resolution of work in flight by the target Db2 must be done in the same mode as the source Db2.

Note: The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

The following error messages are expected. If an in-flight transaction was in progress, multiple messages may be associated with it:

```
DSNI001I  @TGT1 RESTART HAS BEEN DEFERRED
          REASON 00C90095
          TYPE 00000200
          NAME databasename .tablespace or indexspace name
DSNB250E  @TGT1 DSNIIIMPD A PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST
          DATABASE NAME=database name
          SPACE NAME=tablespace or indexspace name
          DATA SET NUMBER=1
          PAGE RANGE X'nnnnnnnn' TO X'nnnnnnnn'
          START LRSN=X'nnnnnnnnnnnnnnnn'
          END LRSN=X'nnnnnnnnnnnnnnnn'
          START RBA=X'nnnnnnnnnnnnnnnn'
```

8. Run the DB2START command again.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem”](#) on page 112 for information on how to set up the special dsnzparm module.

The other target Db2 members may hold locks that need to be released before the DB2FIX command can be run. To release these locks, start the remaining target Db2 members in maintenance mode with the special zparks and then stop the members, as follows:

- Run the DB2START command with the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).
- Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO).

The DB2START and DB2STOP commands can be run in the same job.

The start-up of the target Db2 subsystems must be done in data sharing mode. This is necessary because this is an online cloning and the resolution of work in flight by the target Db2 must be done in the same mode as the source Db2.

Note: The initial starts of the target Db2 subsystems should not be a cold start. A cold start will not allow the target Db2 to resolve in-flight transactions and could leave inconsistencies in Db2 data.

9. Run the DB2FIX command.

Run the DB2FIX command to correct any of the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) page spaces that are restricted.

Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(Db2) on the target subsystem. This will start any page spaces in the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) that have LPL or GRECP status.

Only one Db2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.



Attention: Steps 10, 11, and 12 need to be performed ONLY if the table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and need to be redone. These steps can be automated by using either the DB2FIX return code setting, or the DB2FIX WTO message produced when DB2FIX starts restricted page spaces. For more information, see the DB2FIX command in [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371. Otherwise, proceed to step 13.

10. Run the DB2STOP command.

Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the target Db2 subsystem from running in maintenance mode.

11. Run the DB2UPDATE command again.

Run the DB2UPDATE command with the DBD01ONLY keyword using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary Db2 changes to reflect the renamed data sets. DB2UPDATE with the DBD01ONLY keyword updates only the Db2 directory. The VCATNAME, and optionally, the Db2 storage group names are updated.

The target Db2 XCF structures need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target Db2 XCF structures. If this keyword is set to (N), the target Db2 XCF structures need to be manually deallocated prior to starting the target Db2 subsystem. See [“Deallocating target Db2 coupling facility structures”](#) on page 1160 for more information on this process.

12. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic [“Cloning a Db2 subsystem”](#) on page 112 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target Db2 subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

This start-up of the target Db2 subsystem must be done in data sharing mode.

13. Run the DB2SQL command.

Run the DB2SQL command using SCKZJCL member (CKZDSQL) to update the Db2 catalog. This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic [“Cloning a Db2 subsystem”](#) on page 112.

The DB2SQL command makes the necessary changes to the Db2 catalog. The changes include the VCATNAME, storage group names, volumes, and the GROUP_MEMBER field value in SYSIBM.SYSDATABASE for work databases.

14. Run the DB2FIX command.

Run the DB2FIX command to correct any of the application page spaces that are restricted. Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(APPLICATION) on the target subsystem. This will start any application page spaces that have LPL or GRECP status.

Only one Db2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

15. (Optional) Run the DB2SCHEMA-UPDATE command.

This step assumes that the plan and package have already been installed. To update and re-create objects using new schema values, run the DB2SCHEMA-UPDATE command on the target subsystem using SCKZJCL member (CKZDSUPD). The DB2SCHEMA-UPDATE command changes the creator, owner, and the schema of database objects.

16. Run the DB2STOP command.

To stop the target Db2 subsystem from running in maintenance mode, run the DB2STOP command using SCKZJCL member (CKZDSTO).

17. Run the DB2RBLDBSDS command.

To alter the member BSDS to be non-data sharing, run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for the desired target member using the keyword DATA-SHARING(NO).

18. Run the DB2XCFCLEAN command.

Run the DB2XCFCLEAN command using SCKZJCL member (CKZDXCFC). This step will deallocate the Db2 XCF structures and remove the XCF group members.

19. (Optional) Run the DB2UTILXCLEAN command.

To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source Db2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source Db2 subsystem was cloned.

20. Run the DB2START command.

To start the target Db2 subsystem with its normal (non-data sharing) zparms, run the DB2START command using the REPLY-TO-RESTART-WTOR(Y) keyword on the target subsystem using SCKZJCL member (CKZDSTA). This start-up of the target Db2 subsystem will require a cold start, so an operator reply will need to be made to the Db2 message:

```
DSNJ246I  CONDITIONAL RESTART RECORD INDICATES COLD START AT  
RBA xxxxxxxxxxxx.  REPLY Y TO CONTINUE, N TO CANCEL
```

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command will cause DB2START to automatically reply to the DSNJ246I WTOR message.

Chapter 10. Subsystem cloning using the Db2 Cloning Tool stored procedure

Db2 Cloning Tool provides a Db2 stored procedure that can generate the necessary jobs to perform the subsystem cloning, schedule the jobs in the Db2 administrative task scheduler, and monitor the execution of the jobs. The stored procedure will return to the caller when the requested cloning has ended, either in success or failure.

The cloning may be one of the following:

- From volumes
- From a Db2 system-level backup created by the Db2 BACKUP SYSTEM command (DASD or tape)
- From a user-created system-level backup created by the HSM FRBACKUP command (DASD or tape)
- From data set copies

For data sharing, the target Db2 system may optionally have fewer members or be non-data sharing. The cloning may include one or more non data sharing subsystems or data sharing groups. The Db2 systems that are being cloned must reside on the set of source volumes being used.

A user-created system-level backup can be offline (the source Db2 is down) or online (the source Db2 is active). An online user-created system-level backup is expected to be consistent, and a conditional restart of the target Db2 is not necessary.

About the Db2 Cloning Tool stored procedure

This topic describes the stored procedure inputs and outputs and system requirements for using the stored procedure.

Stored procedure input and output

The input to the stored procedure is:

- The requested operation/type of the run
- The DSN and optional member name of the cloning parameter file

The **cloning parameter file** provides necessary information about the specific cloning. This information includes the job cards, user ID, and password the jobs will run under, the DSN to build the JCL into, the prefix to use for any work file data sets, the source and target volumes, the source and target ICF catalogs, the rename masks, and the source and target Db2 systems. The cloning parameter file contains parameters that identify the product parameter file and Db2 systems parameter file that should be used.

The **product parameter file** provides necessary information about the product. This information includes the DSN of the product load library and the DSN of the CKZINI file.

The **Db2 systems parameter file** provides necessary information about the Db2 systems that will be used by the cloning. This information includes items for each Db2 system, such as the SSID, DSN of the SDSNLOAD library, the system that the Db2 should run on, the system VCAT, and DSNs of the boot strap data sets.

The output from the stored procedure is:

- A return code indicating the success or failure of the requested operation
- A message area containing messages about the success or failure of the requested operation
- A result set containing detail messages

System requirements

The following requirements must be met to use the Db2 Cloning Tool Subsystem Cloning stored procedure:

- The Db2 administrative task scheduler must be configured and available on the Db2 system where the stored procedure is run.
- If the WLM address space the stored procedure runs in does not include the Db2 Cloning Tool load library, the following modules must be copied from the Db2 Cloning Tool load library into a PDSE load library that is available to the WLM address space:
 - CKZ00209
 - CKZ1SPSS
- If the cloning will suspend or stop the source Db2 system, the source Db2 system should not be the same Db2 system that the stored procedure and the Db2 administrative task scheduler are running on. Suspending the Db2 system causes the Db2 administrative task scheduler to freeze, thus preventing the scheduling of the cloning jobs that will resume the source Db2 system. Stopping a Db2 system causes the Db2 administrative task scheduler to stop, thus preventing the scheduling of the cloning jobs that will start the source Db2 system.
- If you want to view the result set that is returned by the stored procedure, ensure that the Tools Customizer step to create the stored procedure global temporary table has been completed as described in the customization topics. For more information, see [“Worksheets: Gathering parameter values for Tools Customizer”](#) on page 31.

Migrating the stored procedure from Db2 Cloning Tool V3.1

These instructions provide a migration path for the V3.1 stored procedure to version V3.2 of Db2 Cloning Tool.

Dropping and re-creating the stored procedure

If you were using the V3.1 stored procedure, you must drop the V3.1 stored procedure and create a new V3.2 stored procedure.

The Tools Customizer steps "Drop Subsystem Cloning stored procedure" and "Define Subsystem Cloning stored procedure" can be used to drop the old stored procedure and create the new one. For more information, see the topics [“Worksheets: Gathering parameter values for Tools Customizer”](#) on page 31 and [Chapter 4, “Customizing Db2 Cloning Tool,”](#) on page 77.

Using the V3.1 parameter list files with V3.2

If you use the stored procedure in Db2 Cloning Tool V3.1, you can use the same parameter list with modifications with V3.2.

In V3.2, the product parameter file SCKZPARM parameter was changed to SCKZEXEC. However, SCKZPARM is accepted for compatibility with V3.1.

In Db2 Cloning Tool V3.2, the stored procedure call parameter list changed. Four parameters have been removed from the call list. These parameter values have been moved into the cloning parameter file.

For Db2 Cloning Tool V3.1, the call statement was as follows:

```
EXEC SQL CALL CKZT00LS.CLONE_SS( :TYPE,  
:PPARMDSN, :PPARMMEM,  
:SPARMDSN, :SPARMMEM,  
:CPARMDSN, :CPARMMEM,  
:SP_RC, :SP_MSGS)
```

For Db2 Cloning Tool V3.2, the call statement is as follows:

```
EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,  
:CPARMDSN, :CPARMMEM,  
:SP_RC, :SP_MSGS)
```

You can use your existing V3.1 stored procedure parameters with modifications. The V3.1 PPARMDSN, PPARMMEM, SPARMDSN, and SPARMMEM values have been moved into parameters in the cloning parameter file as the PPARM-DSN and SPARM-DSN parameters.

For example, the Db2 Cloning Tool V3.1 call statement uses the following:

- PPARMDSN value is CKZ.SP.PARMS
- PPARMMEM value is PRDPARMS
- SPARMDSN value is CKZ.SP.PARMS
- SPARMMEM value is SYSPARMS
- CPARMDSN value is CKZ.SP.PARMS
- CPARMMEM value is CLONE1

For the V3.2 release, you can use your existing cloning parameter files, but add the following parameters to the cloning parameter file CKZ.SP.PARMS(CLONE1):

- PPARM-DSN = CKZ.SP.PARMS(PRDPARMS)
- SPARM-DSN = CKZ.SP.PARMS(SYSPARMS)

Module name changes

For informational purposes, the following module names have changed in this release:

- For V3.1, the modules that are used by the stored procedure were CKZ00200, CKZ00201, CKZ00900, CKZ01DYN, CKZ01HEX, and CKZ01PSN.
- For V3.2, the modules that are used by the stored procedure are CKZ00209 and CKZ1SPSS.

Steps for cloning a Db2 subsystem using the stored procedure

The procedure in this section describes the general steps to clone Db2 subsystems using the stored procedure.

Procedure

1. Set up the product parameter file.
2. Set up the Db2 systems parameter file.
3. Set up the cloning parameter file.
4. Invoke the stored procedure.
5. Verify the cloning.
6. Run the verified cloning.

Parameter files and parameter descriptions

Set up the parameter files that are required for the stored procedure according to the following requirements.

You must create three parameter files that are used by the stored procedure:

- Product parameter file (PPARM)
- Db2 systems parameter file (SPARM)
- Cloning parameter file (CPARM)

Requirements for all parameter files

The three parameter input files must be defined as RECFM=FB, LRECL=80. Each parameter file can be in a sequential data set or can be a member of a PDS or PDSE.

An asterisk (*) in column 1 denotes a comment. Any other non-blank character in column 1 denotes the start of a parameter and is of the format **keyword = value**. A value can consist of multiple items that are separated by one or more blanks. A value can be continued by using a blank in column 1 of the following record. The value that starts in column 2 is concatenated to the prior records column 72.

PPARM parameter file keywords

The product parameter file provides required information about Db2 Cloning Tool. A sample parameter file can be found in SCKZPARM member CKZPPARM. Possible product parameter keywords are:

CKZINI

(Required) The DSN and member that identifies the CKZINI to be used.

SCKZLOAD

(Required) The DSN of the Db2 Cloning Tool load library to be used. Multiple load libraries can be specified.

SCKZEXEC

The DSN of the Db2 Cloning Tool EXEC library to be used. This library contains the CKZRNTGT REXX EXEC. This parameter is required if SOURCE-VOLUMES=DB2SLB or SOURCE-VOLUMES=USERSLB is specified in the CPARMs.

SCKZDBRM

The DSN of the Db2 Cloning Tool DBRM library to be used. This parameter is required if BIND-ON-TARGET=Y is specified.

SPARM parameter file keywords

The Db2 systems parameter (SPARM) file provides required information about the Db2 systems that will be used by the cloning. A sample parameter file can be found in SCKZPARM member CKZSPARM. Possible Db2 system parameter keywords follow. All parameters are optional unless otherwise specified.

SSID

(Required) The SSID of the Db2 subsystem. The SSID keyword denotes the start of each Db2 subsystem entry. The first keyword in the Db2 systems parameter file must be SSID.

SDSNLOAD

(Required) The DSN of the Db2 load library to be used. Multiple load libraries can be specified.

SDSNEXIT

The DSN of the Db2 SDSNEXIT load library to be used. Multiple load libraries can be specified.

BSDS01

(Required for the target Db2) The DSN of the BSDS01 data set to be used.

BSDS02

(Required for the target Db2) The DSN of the BSDS02 data set to be used.

SPECIAL-DSNZPARM

(Required for the target Db2) The name of the special dsnzparm to be used.

NORMAL-DSNZPARM

The name of the dsnzparm to be used for normal starts of the subsystem.

NONDS-DSNZPARM

The name of the dsnzparm to be used when starting the surviving member in non-data sharing mode. Used only when cloning a data sharing group with the target becoming non-data sharing.

EXEC-SYSTEM

The name of the system this Db2 subsystem runs on.

GROUP

The name of the Db2 group this subsystem belongs to. This parameter is required for a data sharing Db2 system.

MEMBER

The member name of this subsystem. This parameter is required for a data sharing Db2 system.

SYSVCAT

(Required for the target Db2) The system VCAT of this subsystem.

DDF-ALIAS

The DDF aliases of this subsystem. This parameter corresponds to the ALIAS keyword of the DB2UPDATE command.

DDF-DYNAMIC-ALIAS

The DDF dynamic location aliases of this subsystem. This parameter corresponds to the DYNAMIC-ALIAS keyword of the DB2UPDATE command.

DDF-GENERIC

The DDF generic luname of this subsystem. This parameter corresponds to the GENERIC keyword of the DB2UPDATE command.

DDF-GRPIPV4

The DDF group IP V4 address for this subsystem. This parameter corresponds to the GRPIPV4 keyword of the DB2UPDATE command.

DDF-GRPIPV6

The DDF group IP V6 address for this subsystem. This parameter corresponds to the GRPIPV6 keyword of the DB2UPDATE command.

DDF-IPNAME

The DDF ipname for this subsystem. This parameter corresponds to the IPNAME keyword of the DB2UPDATE command.

DDF-IPV4

The DDF IP V4 address for this subsystem. This parameter corresponds to the IPV4 keyword of the DB2UPDATE command.

DDF-IPV6

The DDF IP V6 address for this subsystem. This parameter corresponds to the IPV6 keyword of the DB2UPDATE command.

DDF-LOCATION

The DDF location of this subsystem. This parameter corresponds to the LOCATION keyword of the DB2UPDATE command.

DDF-LUNAME

The DDF luname of this subsystem. This parameter corresponds to the LUNAME keyword of the DB2UPDATE command.

DDF-PASSWORD

The DDF password of this subsystem. This parameter corresponds to the PASSWORD keyword of the DB2UPDATE command.

DDF-PORT

The DDF port of this subsystem. This parameter corresponds to the PORT keyword of the DB2UPDATE command.

DDF-RESPORT

The DDF resport of this subsystem. This parameter corresponds to the RESPORT keyword of the DB2UPDATE command.

DDF-SECPORT

The DDF secpport of this subsystem. This parameter corresponds to the SECPORT keyword of the DB2UPDATE command.

CPARM parameter file keywords

The cloning parameter (CPARM) file provides required information about the specific cloning. A sample parameter file can be found in SCKZPARM member CKZCPARM. Possible cloning parameter keywords follow. All parameters are optional unless otherwise specified.

CPARM general parameters

PPARM-DSN

(Required) The data set name (and member name if partitioned) that contains the product parameter file to be used for this cloning. This data set must have attributes of DSORG=PO,RECFM=FB,LRECL=80.

SPARM-DSN

(Required) The data set name (and member name if partitioned) that contains the Db2 systems parameter file to be used for this cloning. This data set must have attributes of DSORG=PO,RECFM=FB,LRECL=80.

JCL-DSN

(Required) The data set name of the library that will contain the JCL for this cloning. This data set must have attributes of DSORG=PO,RECFM=FB,LRECL=80.

STATUS-DSN

(Required) The data set name of the data set that will be used to hold the status of the cloning. This data set must have attributes of DSORG=PS,RECFM=FB,LRECL=200.

TASK-PREFIX

(Required) The prefix that will be used for the task names in the Db2 administrative task scheduler. The maximum length is 16 characters.

USERID

(Required only if not using PassTickets) The user ID to be used by the cloning jobs. If not specified, the user ID that runs the stored procedure with TYPE=BUILD is used. If PASSWORD is not specified, the USERID that is specified must be the same as the user ID that is running the stored procedure when TYPE=BUILD.

PASSWORD

(Required only if not using PassTickets) The password or password phrase of the user ID to be used for the cloning jobs. The maximum allowed length of a password is 8 characters. The maximum allowed length for a password phrase is 24 characters. The password phrase limit is required due to the 24-character limit for the password parameter in the standard Db2 stored procedure, which is used to add a task to the Db2 administrative task scheduler. If not specified, a RACF PasTicket is generated when needed. PASSWORD cannot be specified if USERID is not specified. To generate PassTickets, the stored procedure must run in an APF-authorized environment.

WORK-PREFIX

(Required) The prefix that will be used for the data sets created and used by the cloning jobs. The maximum length is 27 characters.

JOB CARD1 - 9

The job cards that will be used by the cloning jobs. JOBCARD1 is required.

COM01 - 99

Comments about the cloning that will be added to the generated JCL.

CLONING-TYPE

(Required) Specifies the type of cloning. Valid values are OFFLINE or ONLINE.

Note: Cloning from a Db2 system-level backup (SOURCE-VOLUMES=DB2SLB) is an online cloning; in that case, CLONING-TYPE=ONLINE will be used if CLONING-TYPE=OFFLINE has been specified.

BIND-ON-TARGET

Indicates whether a bind of the plan and package should be done on the target Db2 before running DB2SQL. Valid values YES | Y or NO | N. This option addresses the situation where the source Db2 system does not have a current bind of the DB2SQL package; therefore, a bind must be run on the target Db2 before running the DB2SQL command. If YES is specified, the data set that is specified for

the CKZINI keyword in the product parameter file is used to retrieve the DB2PLAN value to be used for generating the bind script. If not specified, the default is NO.

BIND-PACKAGE

Specifies the package name to use in the bind of the plan and package. This parameter applies only when BIND-ON-TARGET is specified. If not specified, a package name of CKZPACK is used.

RESULT-SET

Indicates whether the stored procedure should create a result set to provide details about stored procedure processing. Valid values are YES | Y or NO | N. If not specified, the default is NO.

COPY-BY-DS

Indicates whether the subsystem cloning is by volume (N | NO) or by data set (Y | YES). Valid values are YES | Y or NO | N. If not specified, the cloning is done by volume. If COPY-BY-DS is Y or YES, the SOURCE-VOLUMES keyword is not used.

CPARM COPY command parameters

These parameters are Db2 Cloning Tool Subsystem Cloning COPY or RESTORE-FROM-DUMPTAPES command parameters. More information about these parameters is available in the topic [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371.

KEEP-VOLUMES-SEQUENCE

Specifies that source volumes are paired with target volumes strictly by sequence number. This parameter corresponds to the KEEP-VOLUMES-SEQUENCE keyword of the COPY and RESTORE-FROM-DUMPTAPES commands. For the COPY command, the source volumes that are defined in the FROM-STORAGEGROUP parameter are copied to the target volumes that are defined in the TO-STORAGEGROUP parameter in the same position. For the RESTORE-FROM-DUMPTAPES command, the source volumes that are defined in the SOURCE-STORAGEGROUP parameter are copied to the target volumes that are defined in the TO-STORAGEGROUP parameter in the same position. For the COPY command, the source volumes that defined in the FROM-VOLSER parameter are copied to the target volumes that are defined in the TO-VOLSER parameter in the same position. For additional information about how the pairing is accomplished, see [“COPY command and keyword definitions”](#) on page 388 or [“RESTORE-FROM-DUMPTAPES command and keyword definitions”](#) on page 491. Valid values are YES | Y or NO | N.

FROM-STORAGEGROUP

The source storage groups of the volume copy. This parameter is required if FROM-VOLSER and SOURCE-VOLUMES are not specified. This parameter is mutually exclusive with SOURCE-VOLUMES. This parameter corresponds to the FROM-STORAGEGROUP keyword of the COPY command.

TO-STORAGEGROUP

The target storage groups of the volume copy or restore from system-level backup dump tapes. This parameter is required if TO-VOLSER is not specified. This parameter corresponds to the TO-STORAGEGROUP keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

FROM-VOLSER

The source volumes of the volume copy. This parameter is required if FROM-STORAGEGROUP and SOURCE-VOLUMES are not specified. This parameter is mutually exclusive with SOURCE-VOLUMES.) This parameter corresponds to the FROM-VOLSER keyword of the COPY command.

TO-VOLSER

The target volumes of the volume copy or restore from system-level backup dump tapes. This parameter is required if TO-STORAGEGROUP is not specified. This parameter corresponds to the TO-VOLSER keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

EXCLUDE-FROM-VOLSER

The volumes, or volumes that match a mask, to be excluded from being specified for either the FROM-VOLSER parameter or the FROM-STORAGEGROUP parameter. This parameter corresponds to the EXCLUDE-FROM-VOLSER keyword of the COPY command.

EXCLUDE-TO-VOLSER

The volumes, or volumes that match a mask, to be excluded (not selected as targets) from being specified for either the TO-VOLSER parameter or the TO-STORAGEGROUP parameter. This parameter

corresponds to the EXCLUDE-TO-VOLSER keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

SOURCE-VOLUMES

Specifies that the cloning source volumes are from a system-level backup. Valid values for this parameter are DB2SLB (the source of the cloning is a Db2 SLB created by a Db2 BACKUP SYSTEM) and USERSLB (the source of the cloning is a user-created SLB created by an HSM FRBACKUP). This parameter is mutually exclusive with FROM-STORAGEGROUP and FROM-VOLSER.

SOURCE-STORAGEGROUP

Specifies the source storage group names to be used when pairing the backup volumes to the target volumes. This parameter is optional, and will only be used if SOURCE-VOLUMES=DB2SLB or SOURCE-VOLUMES=USERSLB is specified. The values that are specified for this parameter are used for building a FROM-USER-STORAGEGROUP keyword in a COPY command or a SOURCE-STORAGEGROUP keyword in a RESTORE-FROM-DUMPTAPES command.

SOURCE-LOCATION

Specifies the location to use to determine the cloning source volumes from a system-level backup (SLB). This parameter is required if SOURCE-VOLUMES=DB2SLB or SOURCE-VOLUMES=USERSLB is specified. This parameter is mutually exclusive with SOURCE-DB-COPYPOOL-NAME and SOURCE-LG-COPYPOOL-NAME. This parameter corresponds to the LOCATION keyword of the DB2GETBACKINFO command.

SOURCE-DB-COPYPOOL-NAME

Specifies the database copypool name to use to determine the cloning source volumes from a system-level backup (SLB). This parameter is required if SOURCE-VOLUMES=DB2SLB or SOURCE-VOLUMES=USERSLB is specified. This parameter is mutually exclusive with SOURCE-LOCATION. If SOURCE-DB-COPYPOOL-NAME is specified, SOURCE-LG-COPYPOOL-NAME must also be specified. This parameter corresponds to the DB-COPYPOOL-NAME keyword of the DB2GETBACKINFO command.

SOURCE-LG-COPYPOOL-NAME

Specifies the log copypool name to use to determine the cloning source volumes from a system-level backup (SLB). This parameter is required if SOURCE-VOLUMES=DB2SLB or SOURCE-VOLUMES=USERSLB is specified. This parameter is mutually exclusive with SOURCE-LOCATION. If SOURCE-LG-COPYPOOL-NAME is specified, SOURCE-DB-COPYPOOL-NAME must also be specified. This parameter corresponds to the LG-COPYPOOL-NAME keyword of the DB2GETBACKINFO command.

SOURCE-TOKEN

Specifies the token of the backup to be used. A value of LAST uses the latest (most recent) backup. This parameter is required if SOURCE-VOLUMES=DB2SLB or SOURCE-VOLUMES=USERSLB is specified. This parameter corresponds to the TOKEN and LAST keywords of the DB2GETBACKINFO command.

SOURCE-USE-DUMPTAPES

Specifies that system-level backup dump tapes are used for the cloning. This parameter is used only if SOURCE-VOLUMES= DB2SLB or SOURCE-VOLUMES=USERSLB is specified. Valid values are YES | Y and NO | N. If not specified, the default is NO.

SOURCE-DUMP-CLASS

Specifies the dump class of the system-level backup dump tapes to be used for the cloning. This parameter is used only if SOURCE-USE-DUMPTAPES = Y or YES is specified. This parameter corresponds to the DUMP-CLASS keyword of the DB2GETBACKINFO command.

SOURCE-BACKINFO-DSN

Specifies the data set name to use for the BACKINFO file. This parameter is used for a cross-sysplex system-level backup cloning. The BACKINFO data set is created by a DB2GETBACKINFO run on the source system and transferred to the target system, where it will be used by the cloning. The DB2GETBACKINFO command is not executed on the target system as part of the cloning. This parameter is used only if SOURCE-VOLUMES= DB2SLB or SOURCE-VOLUMES=USERSLB is specified.

TARGET-UCATS-ON-TARGET-VOLUMES

Specifies that there are target ICF catalogs that reside on the target volumes, and you want the target ICF catalogs to remain on the target volumes after the RENAME step. If Y is specified, when the RENAME step completes, the target ICF catalogs will reside on target volumes.

USERCATALOGS

(Required) Specifies the source ICF catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in. This parameter corresponds to the USERCATALOGS keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

DM-PGM

Specifies the data mover program to be used for volume copies. Valid values are ADRDSSU or EMCSNAP. This parameter corresponds to the PGM subkeyword of the DATA-MOVER keyword of the COPY command. If DM-PGM is not specified, ADRDSSU is used for volume copies.

DM-BACKGROUND COPY

Indicates whether background copy should be done for the volume copies. This parameter applies only when DM-PGM=EMCSNAP is specified. This parameter corresponds to the BACKGROUND COPY subkeyword of the DATA-MOVER keyword of the COPY command.

DM-CHECKONLINEPATHSTATUS

Indicates that before performing a volume snap, a check will be conducted to ensure paths from other CPUs to the target devices are offline. This parameter applies only when DM-PGM=EMCSNAP is specified. This parameter corresponds to the CHECKONLINEPATHSTATUS subkeyword of the DATA-MOVER keyword of the COPY command.

DM-CHECKVTOC

Specifies whether a VTOC analysis of the source volume is performed during copy processing. Valid values are YES | Y (generate the DSS CHECKVTOC keyword) or NO | N (do not generate the DSS CHECKVTOC keyword). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. This parameter corresponds to the CHECKVTOC subkeyword of the DATA-MOVER keyword of the COPY command.

DM-CONSISTENT

Indicates one of the following:

- If DM-PGM is not specified or DM-PGM=ADRDSSU is specified, indicates whether to use a FlashCopy Consistency Group. Valid values are YES | Y and NO | N.
- If DM-PGM=EMCSNAP is specified, indicates whether to use Enginuity Consistency Assist (ECA) for consistent SNAP VOLUME operations. Valid values are YES | Y and NO | N.

This parameter corresponds to the CONSISTENT subkeyword of the DATA-MOVER keyword of the COPY command.

DM-COPYCMDLIMIT

Specifies the maximum COPY FULL commands that are built by Db2 Cloning Tool for each DSS execution. This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. This parameter corresponds to the COPYCMDLIMIT subkeyword of the DATA-MOVER keyword of the COPY command.

DM-DIFFERENTIAL

Indicates whether to use the Enginuity Differential Snap feature for SNAP VOLUME operations. This parameter applies only when DM-PGM=EMCSNAP is specified.

DM-FASTREP

Indicates whether fast replication is preferred (PREF), required (REQ), or not required (NONE). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. This parameter corresponds to the FASTREP subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCNOCOPY

Indicates whether background copy should be done for the volume copies. Valid values are YES | Y (do not use background copy) or NO | N (use background copy). This parameter applies only when

DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the FCNOCOPY subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCSETGTOK

Indicates whether a FlashCopy target volume can also be a space efficient volume. Valid values are YES | Y (the target volumes can be space efficient volumes; the FCSETGTOK keyword will be generated) or NO | N (the target volumes cannot be space efficient volumes; the FCSETGTOK keyword will not be generated). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the FCSETGTOK subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCTOPPRCPRIARY

Indicates whether a FlashCopy target volume can also be a PPRC primary volume. Valid values are:

- YES | Y: The target volumes can be PPRC primary volumes. The FCTOPPRCPRIARY keyword will be generated.
- NO | N: The target volumes cannot be PPRC primary volumes. The FCTOPPRCPRIARY keyword will not be generated.
- PRESMIRREQ | PMR: The FCTOPPRCPRIARY(PRESMIRREQ) keyword will be generated.
- PRESMIRPREF | PMP: The FCTOPPRCPRIARY(PRESMIRPREF) keyword will be generated.
- PRESMIRNONE | PMN: The FCTOPPRCPRIARY(PRESMIRNONE) keyword will be generated.

This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the FCTOPPRCPRIARY subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCTOXRCPRIARY

Indicates whether a FlashCopy target volume can also be an XRC primary volume. Valid values are:

- YES | Y: The target volumes can be XRC primary volumes. The FCTOXRCPRIARY keyword will be generated.
- NO | N: The target volumes cannot be XRC primary volumes. The FCTOXRCPRIARY keyword will not be generated.

This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. Do not specify the FCTOXRCPRIARY keyword with the DM-FASTREP=NONE keyword. This parameter corresponds to the FCTOXRCPRIARY keyword of the DATA-MOVER keyword of the COPY-BY-DS command.

DM-INCREMENTAL

Indicates whether a full volume Incremental FlashCopy relationship is to be established. This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the INCREMENTAL subkeyword of the DATA-MOVER keyword of the COPY command.

DM-MAXIMUM-SUBTASKS

Sets the absolute maximum number of subtasks that can be attached and used for volume snaps. This parameter applies only when DM-PGM=EMCSNAP is specified. This parameter corresponds to the MAXIMUM-SUBTASKS subkeyword of the DATA-MOVER keyword of the COPY command.

DM-NOCONCURRENT

Specifies whether the CONCURRENT option will be supplied to ADRDUSSU. Valid values are YES | Y (the CONCURRENT option will not be supplied to ADRDUSSU; the NOCONCURRENT keyword will be generated) or NO | N (the CONCURRENT option will be supplied to ADRDUSSU; the NOCONCURRENT keyword will not be generated). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the NOCONCURRENT subkeyword of the DATA-MOVER keyword of the COPY command.

CPARM COPY-BY-DS command parameters

These parameters are Db2 Cloning Tool Subsystem Cloning COPY-BY-DS command parameters. More information about these parameters is available in the topic [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371. These parameters are used when COPY-BY-DS = Y or YES is specified.

REPLACE-UNCONDITIONAL

Specifies whether existing target data sets can be replaced by the copy. Valid values are YES | Y (replace existing target data sets) or NO | N (do not replace target data sets). If not specified, the default is that no existing target data sets can be replaced by the copy.

MIGRATED-DSN

Specifies whether migrated data sets will produce an error (ERROR) or be skipped (SKIP) by the COPY-BY-DS command.

NUM-DS

Specifies the estimated number of total data sets per RENAME-MASKS mask that the COPY-BY-DS command will process. Valid values are 1 through 99999. If not specified, the default is 10000.

DM-DSNS-PER-COPY

The number of data sets to include in each COPY statement. Valid values are 1 through 255. Values that are too high may cause resource shortage errors, particularly with the TIOT. This number interacts directly with DM-DSS-COPY-COMMANDS; the recommended maximum for these two values multiplied together is 1400 or less.

DM-DSS-COPY-COMMANDS

The number of COPY commands to pass in each unique call to ADRDSSU. Valid values are 1 through 10. Values that are too high may cause resource shortage errors, particularly with the TIOT. This number interacts directly with DM-DSNS-PER-COPY; the recommended maximum for these two values multiplied together is 1400 or less.

DM-FCTOPPRCPRIMARY

Indicates whether a FlashCopy target volume can also be a PPRC primary volume. Valid values are:

- YES | Y: The target volumes can be PPRC primary volumes. The FCTOPPRCPRIMARY keyword will be generated
- NO | N: The target volumes cannot be PPRC primary volumes. The FCTOPPRCPRIMARY keyword will not be generated.
- PRESMIRREQ | PMR: The FCTOPPRCPRIMARY(PRESMIRREQ) keyword will be generated.
- PRESMIRPREF | PMP: The FCTOPPRCPRIMARY(PRESMIRPREF) keyword will be generated.
- PRESMIRNONE | PMN: The FCTOPPRCPRIMARY(PRESMIRNONE) keyword will be generated.

This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. Do not specify the FCTOPPRCPRIMARY keyword with the DM-FASTREP=NONE keyword. This parameter corresponds to the FCTOPPRCPRIMARY keyword of the DATA-MOVER keyword of the COPY-BY-DS command.

COPY-BY-DS-MGMTCLAS

Specifies the management class that replaces the source data set management class as input to the ACS routines. COPY-BY-DS-MGMTCLAS can have only one value that must be 1 to 8 characters. COPY-BY-DS-MGMTCLAS is mutually exclusive with COPY-BY-DS-NULLMGMTCLAS.

COPY-BY-DS-NULLMGMTCLAS

Specifies that the input to the ACS routines is a null management class rather than the source data set's management class. Valid values are YES|Y or NO|N. COPY-BY-DS-NULLMGMTCLAS is mutually exclusive with COPY-BY-DS-MGMTCLAS.

COPY-BY-DS-STORCLAS

Specifies the storage class that replaces the source storage class as input to the ACS routines. COPY-BY-DS-STORCLAS can have only one value that must be 1 to 8 characters. COPY-BY-DS-STORCLAS is mutually exclusive with COPY-BY-DS-NULLSTORCLAS.

COPY-BY-DS-NULLSTORCLAS

Specifies that the input to the ACS routines is to be a null storage class rather than the source data set's storage class. Valid values are YES|Y or NO|N. COPY-BY-DS-NULLSTORCLAS is mutually exclusive with COPY-BY-DS-STORCLAS.

CPARM RENAME parameters

These parameters are Db2 Cloning Tool Subsystem Cloning RENAME command parameters. More information about these parameters is available in topic [Chapter 26, "Db2 Cloning Tool Subsystem Cloning commands,"](#) on page 371.

NOTRENAMED

Specifies the NOTRENAMED value to be used in the rename.

ISSUE-UCAT-UNALLOCATE

Specifies the ISSUE-UCAT-UNALLOCATE value to be used in the rename.

ISSUE-VCLOSE

Specifies the ISSUE-VCLOSE value to be used in the rename.

MAX-TASKS

Specifies the MAX-TASKS value to be used in the rename.

DATACLAS

Specifies the SMS DATACLAS to be used for all renamed data sets on SMS managed volumes.

DATACLAS-PAIRS

Specifies source/target pairs for DATACLAS.

DRIVEACS

Specifies that SMS class information for renamed data sets is to be derived by driving ACS routines. Possible values are YES | Y (generated the DRIVEACS keyword) or NO | N (do not generate the DRIVEACS keyword).

GDG-ALL-MIGRATED

Specifies the action to take if all the source GDG generations have been migrated.

GDG-EMPTY

Specifies the action to take if the source GDG base is empty.

GDG-MIGRATED

Specifies the action to take if some, but not all, of the source GDG generations have been migrated.

GDG-TAPE

Specifies the action to take if some, but not all, of the source GDG generations are on tape.

MISSINGUCAT

Specifies the disposition and return code to be generated for data sets found on a volume, where the data set name matches a rename mask, but the catalog back-pointer is not one of the source catalogs that are specified in the corresponding COPY command.

MGMTCLAS

Specifies the SMS MGMTCLAS to be used for all renamed data sets on SMS managed volumes.

MGMTCLAS-PAIRS

Specifies source/target pairs for MGMTCLAS.

ORPHANCATENTRY

Specifies the disposition and return code to be generated for data sets found in a catalog but one or more catalog volume cells are not in the list of volumes copied.

RECATALOG

Specifies that Db2 Cloning Tool can replace an existing catalog entry without considering it an error.

STORCLAS

Specifies the SMS STORCLAS to be used for all renamed data sets on SMS managed volumes.

STORCLAS-PAIRS

Specifies source/target pairs for STORCLAS.

RENAME-MASKS

(Required) Specifies the rename masks that will be used to rename the data sets. (required)

EXCLUDE-SRCNAME-MASKS

This parameter supplies a list of source data set names or masks that will not be renamed (excluded from renaming).

EXCLUDE-SRCNAME

This parameter specifies the return code that will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword.

RENAME-ERROR

This option specifies how processing proceeds when a RENAME error is encountered.

RENAME-AUDIT-LOG

Specifies whether an audit log of the data sets being renamed is to be created by RENAME volume processing.

RENAME-LIST

Specifies whether a list of the renamed data sets is to be produced by RENAME volume processing.

TEMPDSN

Specifies the disposition of temporary data sets and the return code to be generated if at least one occurrence is discovered.

UPDATE-IAM-ASSOCIATIONS

Specifies whether IAM data set associations are to be updated as part of RENAME processing.

VALIDATE-SMS-CLASSES

Specifies whether the SMS class names that are specified in the DATACLAS, DATACLAS-PAIRS, MGMTCLAS, MGMTCLAS-PAIRS, STORCLAS, and STORCLAS-PAIRS keywords will be validated as being defined to SMS.

CPARM BCSCLEAN command parameters

These parameters are Db2 Cloning Tool Subsystem Cloning BCSCLEAN command parameters. More information about these parameters is available in the topic [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371.

CLEANUP-CATALOG-ORPHANS

Specifies whether BCSCLEAN should remove target catalog entries that belong to data sets that are on the target volumes but were not put there by the prior cloning. Valid values are YES | Y (generate the CLEANUP-CATALOG-ORPHANS keyword) or NO | N (do not generate the CLEANUP-CATALOG-ORPHANS keyword). If not specified, the default is NO.

CLEANUP-CATALOG-DSNMASKS

This parameter supplies a list of data set names or masks of data sets that are not on any target volume, but are in a target user catalog and that should be deleted by BCSCLEAN.

CPARM RESTORE-FROM-DUMPTAPES command parameters

These parameters are Db2 Cloning Tool Subsystem Cloning RESTORE-FROM-DUMPTAPES command parameters. More information about these parameters is available in the topic [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371.

RESTORE-DUMP-TAPE-UNIT

Specifies the tape unit to use for allocating the tape drives for the volume restores.

RESTORE-DUMP-MAX-TAPEDRIVES

Specifies the maximum number of tape drives to use for the volume restores. Valid values are 1-16, or specify an asterisk (*) to allow the RESTORE-FROM-DUMPTAPES command to set the maximum (set to 4 for this release).

RESTORE-DUMP-VARY-SCOPE

Specifies the scope of the vary commands against the target volumes. Valid values are GLOBAL or LOCAL.

RESTORE-DUMP-WAIT-TAPE-ALLOC

Specifies parameters for the WAIT-TAPE-ALLOC(*nnn*, RC(*rrrr*)) keyword. If RESTORE-DUMP-WAIT-TAPE-ALLOC is provided, specify the *nnn* value as the number of minutes in the range 1-999, and the *rrrr* value as the return code in the range 8-4095.

CPARM parameters for pairs of subsystems or data sharing groups

The DB2-SUBSYSTEM-PAIR and DB2-GROUP-PAIR keywords denote the start of the cloning parameters for a pair of subsystems or data sharing groups.

The DB2-SUBSYSTEM-PAIR keyword denotes a non-data sharing pair of a source Db2 SSID and a target Db2 SSID.

The DB2-GROUP-PAIR keyword denotes a data sharing group pair of a source Db2 group and a target Db2 group. Use the SSID-PAIRS keyword to specify the source Db2 SSID and target Db2 SSID pairs of the members of the data sharing group. Also, specify the DATASHR-ATTR keyword to set the final data sharing attributes for the target group. In addition, you can specify the SSID-SURVIVING keyword to identify the target Db2 SSIDs that will survive for DATASHR-ATTR of FEWER or NONDS. For NONDS, there can be only one surviving SSID.

DB2-SUBSYSTEM-PAIR

Specifies the source Db2 SSID and its corresponding target Db2 SSID. This keyword marks the start of the keywords for a subsystem pair. This parameter is required for each non-data sharing Db2 being cloned.

DB2-GROUP-PAIR

Specifies the source Db2 group name and its corresponding target Db2 group name. This keyword marks the start of the keywords for a group pair. This parameter is required for each data sharing group that is being cloned.

SSID-PAIRS

Specifies the pairs of the source Db2 SSIDs and their corresponding target Db2 SSIDs. This parameter is required for DB2-GROUP-PAIR.

DATASHR-ATTR

Specifies the data sharing attribute that the final target Db2 system will have. Valid values are SAME, FEWER, or NONDS. This parameter is required for DB2-GROUP-PAIR.

SSID-SURVIVING

Specifies the surviving target SSIDs. This parameter is required when DATASHR-ATTR=FEWER or NONDS is specified and applies only when DATASHR-ATTR is FEWER or NONDS.

DB2-HLQS

(Required) Specifies the source and target Db2 data set high level qualifiers. This parameter corresponds to the DB2-HLQS keyword of the DB2UPDATE command.

STOGROUPS

Specifies the source and target storage groups that are defined to the Db2 subsystem. This parameter corresponds to the STOGROUPS keyword of the DB2UPDATE command.

ARCHIVE

If the Db2 archive logs are on the source DASD volumes, and are copied to the target volumes, this keyword specifies whether the data set names and VOLSERS of the Db2 archive logs in the BSDSs are to be changed to the target values. Valid values are YES | Y (generate the ARCHIVE keyword) or NO | N (do not generate the ARCHIVE keyword). This parameter corresponds to the ARCHIVE keyword of the DB2UPDATE command.

LISTSQL

Specifies whether the generated SQL should be included in the listing of the DB2SQL command. This parameter corresponds to the LISTSQL keyword of the DB2SQL command.

WLM-ENVIRONMENT-MASKS

Specifies the masks that will be used to rename the WLM environment names in SYSIBM.SYSROUTINES. This parameter corresponds to the WLM-ENVIRONMENT-MASKS keyword of the DB2SQL command.

DATACLAS-MASKS

Specifies the masks that will be used to rename the DATACLAS values in SYSIBM.SYSSTOGROUP. This parameter corresponds to the DATACLAS-MASKS keyword of the DB2SQL command.

MGMTCLAS-MASKS

Specifies the masks that will be used to rename the MGMTCLAS values in SYSIBM.SYSSTOGROUP. This parameter corresponds to the MGMTCLAS-MASKS keyword of the DB2SQL command.

STORCLAS-MASKS

Specifies the masks that will be used to rename the STORCLAS values in SYSIBM.SYSSTOGROUP. This parameter corresponds to the STORCLAS-MASKS keyword of the DB2SQL command.

CLEAN-SYSUTILX

Specifies whether SYSUTILX is to be cleaned out as part of the cloning. The cleaning out of SYSUTILX is done by using a job with the DB2UTILXCLEAN command. Valid values are YES | Y (generate the DB2UTILXCLEAN job) or NO | N (do not generate the DB2UTILXCLEAN job). If this parameter is not specified, the default is NO (to not clean out SYSUTILX).

EXECUTE-CREATE-DDL

Specifies whether the DDL that is generated in the CREATE-DDL-DDN DD is run. Valid values are YES | Y (execute the created DDL) or NO | N (do not execute). If this parameter is not specified, the default is YES. Specify NO | N if a different DDL job should be run, or if the dropped objects are not needed after the cloning. This parameter corresponds to the EXECUTE-CREATE-DDL keyword of the DB2SCHEMA-UPDATE command.

USE-DDL-SQLID

Specifies the SQLID to be used in the CURRENT SQLID parameter of DDL statements for the DB2SCHEMA-UPDATE command. The SQLID that is provided can be up to eight characters long. This parameter corresponds to the USE-DDL-SQLID keyword of the DB2SCHEMA-UPDATE command.

DB2FIX-EXCLUDE-MASKS

Specifies the exclude masks to be used in the DB2FIX DATABASES(APPLICATION) job. This parameter corresponds to the EXCLUDE-MASKS keyword of the DB2FIX command.

DB2FIX-MAX-CONCURRENT-CMDS

Specifies the maximum number of start commands that will be processed concurrently in the DB2FIX DATABASES(APPLICATION) job. This parameter corresponds to the MAX-CONCURRENT-CMDS keyword of the DB2FIX command.

SCHEMA-MASKS

Specifies the source and target schemas in *oldvalue newvalue* pairs, to be used to generate a DB2SCHEMA-UPDATE job. This parameter corresponds to the SCHEMA-MASKS keyword of the DB2SCHEMA-UPDATE command.

Calling the stored procedure

This topic details the syntax for calling the Db2 Cloning Tool stored procedure.

Use the following SQL CALL statement to run the stored procedure:

```
EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,  
    :CPARMDSN, :CPARMMEM,  
    :SP_RC, :SP_MSGS)
```

The input parameters are:

TYPE

Refer to the following section for more information on this parameter; 10 characters are allowed.

CPARMDSN

The fully qualified data set name for the cloning parameter file; 44 characters are allowed.

CPARMMEM

If the cloning parameter file is a PDS, the member name that contains the cloning parameters; 8 characters are allowed.

The parameter files and their contents are described in detail in [“Parameter files and parameter descriptions” on page 155](#).

The output parameters are:

SP_RC

Returns an integer that represents the return code of the cloning job.

SP_MSGS

Contains one or more messages describing the outcome of the cloning job, in varchar format (length 1,331 bytes.)

TYPE parameters

The TYPE parameter identifies what function the stored procedure is to perform. It must be one of the following:

BUILDJCL

Builds JCL only.

BUILD

Builds JCL and adds tasks to the Db2 administrative task scheduler. Sets up the environment for CLONE, RECLONE, and CLEAN.

CLONE

Runs the initial cloning.

RECLONE

Stops the target Db2 systems, runs BCSCLEAN to clean up from the previous cloning and then runs the cloning.

REMOVE

Deletes all JCL and removes tasks from the Db2 administrative task scheduler. If CLONE or RECLONE have been done, a CLEAN should be done before REMOVE.

CLEAN

Stops the target Db2 systems and runs BCSCLEAN to clean up from the previous cloning. Can be used when the clone will no longer be used.

Using a REXX EXEC to call the stored procedure

The stored procedure can be called by using a REXX EXEC.

You can use Tools Customizer to set up a customized REXX EXEC to call the stored procedure, and to set up a job to run the REXX EXEC. For more information about setting up the REXX EXEC and the associated job, see [“Task: Subsystem Cloning tasks” on page 48](#) in the [“Worksheets: Gathering parameter values for Tools Customizer” on page 31](#) topic. The default name of the generated REXX EXEC is CKZCSPSS.

Use the following syntax to run the REXX EXEC:

```
CKZCSPSS type ssid cloning-parm-dsn cloning-parm-member
```

Where:

- CKZCSPSS is the name of the REXX EXEC.
- *type* is the requested function: BUILDJCL, BUILD, CLONE, RECLONE, CLEAN, or REMOVE.
- *ssid* is the Db2 SSID where the stored procedure is to run.
- *cloning-parm-dsn* is the fully qualified data set name for the cloning parameter file.
- *cloning-parm-mbr* is the member name that contains the cloning parameters, if the cloning parameter file is a PDS.

A sample job that invokes the subsystem cloning stored procedure by using a REXX EXEC is shown in the following figure:

```

//jobcard
/*JOBPARM S=SYSA
//*****
//* INVOKE THE SUBSYSTEM CLONING STORED PROCEDURE USING THE *
//* REXX EXEC. *
//* *
//* COMMAND SYNTAX: *
//* CKZCSPSS TYPE SSID CPDSN CPMBR *
//* WHERE: *
//* TYPE? IS THE CLONING FUNCTION TYPE *
//* DB2 ssid *
//* CPDSN? IS THE CLONING PARAMETERS FILE DSN *
//* CPMBR? IS THE MEMBER NAME OF THE CLONING PARAMETERS *
//* *
//*****
//JOBLIB DD DISP=SHR,
// DSN=DB2.SDSNEXIT
// DD DISP=SHR,
// DSN=DB2.SDSNLOAD
// DD DISP=SHR,
// DSN=DB2.RUNLIB.LOAD
//*
//S1 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSEXEC DD DISP=SHR,
// DSN=DB2T00L.CKZ.EXEC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CKZCSPSS TYPE? DB1P CPDSN? CPMBR?
//*
//

```

Figure 11. Sample job to invoke the subsystem cloning stored procedure by using a REXX EXEC

Verifying the cloning

Before performing the cloning, you should verify that the cloning will produce the desired results. Follow the steps in this topic to verify the cloning.

Procedure

1. Invoke the stored procedure with TYPE = BUILDJCL.
2. Check the stored procedure output. If SP_RC for BUILDJCL is not zero, use the messages in SP_MSGS to determine the changes that need to be made. Invoke the stored procedure with TYPE=REMOVE (depending on the BUILDJCL error, REMOVE may produce a non-zero return code), make changes to the parameter files, and then invoke the stored procedure with TYPE=BUILDJCL.
3. Once SP_RC for BUILDJCL equals zero, review the members in the JCL library (JCL-DSN parameter) to ensure that the generated jobs look as expected. If the generated jobs do not look as expected, invoke the stored procedure with TYPE=REMOVE, make changes to the parameter files, and then invoke the stored procedure with TYPE=BUILDJCL.
4. Once the generated JCL looks as expected, manually run the jobs in member name sequence.

Results

Once the jobs run successfully and produce the desired target Db2 system, the cloning has been verified.

JCL members generated by the stored procedure

This topic describes the JCL members generated by the Db2 Cloning Tool stored procedure.

The members put into the JCL library (JCL-DSN parameter) have a form of STnnn, where *nnn* starts at 001 and increments for each additional member.

- The first JCL library member(s) starting with ST001 are a DB2STOP of the target Db2 subsystem(s).
- Following the DB2STOP member(s) is a JCL library member for BCSCLEAN.
- Following the BCSCLEAN member are the cloning members, including DB2STOP/DB2SETLOG SUSPEND, COPY, DB2START/DB2SETLOG RESUME, and RENAME.

TYPE= CLEAN runs the members starting from ST001 (DB2STOP) through BCSCLEAN.

TYPE = CLONE runs the members starting after BCSCLEAN.

TYPE = RECLONE runs all the members.

For example, an online cloning of a typical Db2 subsystem generates the members in the following table:

Table 41. Stored procedure generated members for an online cloning

| Member name | Description | Run for TYPE= CLEAN | Run for TYPE= CLONE | Run for TYPE= RECLONE |
|-------------|--------------------------------|---------------------|---------------------|-----------------------|
| ST001 | DB2STOP target Db2 | Yes | No | Yes |
| ST002 | BCSCLEAN | Yes | No | Yes |
| ST003 | DB2SETLOG SUSPEND source Db2 | No | Yes | Yes |
| ST004 | COPY | No | Yes | Yes |
| ST005 | DB2SETLOG RESUME source Db2 | No | Yes | Yes |
| ST006 | RENAME | No | Yes | Yes |
| ST007 | DB2UPDATE target Db2 | No | Yes | Yes |
| ST008 | DB2START SPECIAL target Db2 | No | Yes | Yes |
| ST009 | DB2FIX DATABASES (Db2) | No | Yes | Yes |
| ST010 | DB2SQL | No | Yes | Yes |
| ST011 | DB2FIX DATABASES (APPLICATION) | No | Yes | Yes |
| ST012 | DB2SCHEMA-UPDATE | No | Yes | Yes |
| ST013 | DB2STOP target Db2 | No | Yes | Yes |
| ST014 | DB2START NORMAL target Db2 | No | Yes | Yes |

Building and running the verified cloning

Once the cloning has been verified, you must build the verified cloning, and then submit the jobs to clone the subsystem.

Building the verified cloning

1. Invoke the stored procedure with TYPE = BUILD.
2. If SP_RC equals zero, the build has completed and the cloning is ready to run. If SP_RC is not zero, use the messages in SP_MSGS to determine the problem.

Running the verified cloning

1. Invoke the stored procedure with TYPE = CLONE.
2. If SP_RC equals zero, the cloning has completed and the target Db2 system should be running. If SP_RC is not zero, use the messages in SP_MSGS to determine the problem. It might be necessary to look at the output of the cloning jobs that have run to determine the problem.

If you must change the generated JCL:

1. Invoke the stored procedure with TYPE=REMOVE.
2. Make changes to the parameter files.
3. Invoke the stored procedure with TYPE=BUILD.

Stored procedure results set

The stored procedure can optionally provide a results set that provides additional details about its processing.

The result set is returned in a global temporary table. The table is defined using JCL generated by Tools Customizer. For more information about defining the global temporary table, see [“Task: Subsystem Cloning tasks” on page 48](#) in the [“Worksheets: Gathering parameter values for Tools Customizer” on page 31](#) topic.

The result set is formatted as shown in the following table:

| Column name | Data type | Contents |
|-------------|--------------|----------------------------|
| SEQNO | INTEGER | Sequence number of the row |
| TEXT | VARCHAR(133) | Detail line |

To enable the results from the stored procedure to be returned via the global temporary table, add the following parameter to the cloning parameter file:

```
RESULT-SET = YES
```

If you include this parameter and the result set is successfully returned, the call to the stored procedure returns an SQLCODE of +466.

Use the following example SQL statements to view the results set:

```
EXEC SQL ASSOCIATE LOCATOR(:result)
      WITH PROCEDURE CKZ.CLONE_SS
EXEC SQL ALLOCATE C101 CURSOR FOR RESULT SET :result
EXEC SQL FETCH C101 INTO :seqno, :text
```

The FETCH is executed until the SQLCODE is not 0.

The following example shows the TEXT field portion of a result set that is returned:

```
CKZ.SP.JCL(ST001) created - taskname: CLONE1_ST001 - description: DB2STOP RC0 for Db2 Cloning Tool
CKZ.SP.JCL(ST002) created - taskname: CLONE1_ST002 - description: BCSCLEAN for Db2 Cloning Tool
CKZ.SP.JCL(ST003) created - taskname: CLONE1_ST003 - description: DB2SETLOG SUSPEND for Db2 Cloning Tool
CKZ.SP.JCL(ST004) created - taskname: CLONE1_ST004 - description: COPY for Db2 Cloning Tool
CKZ.SP.JCL(ST005) created - taskname: CLONE1_ST005 - description: DB2SETLOG RESUME for Db2 Cloning Tool
CKZ.SP.JCL(ST006) created - taskname: CLONE1_ST006 - description: RENAME for Db2 Cloning Tool
CKZ.SP.JCL(ST007) created - taskname: CLONE1_ST007 - description: DB2UPDATE for Db2 Cloning Tool
CKZ.SP.JCL(ST008) created - taskname: CLONE1_ST008 - description: DB2START SPECIAL for Db2 Cloning Tool
CKZ.SP.JCL(ST009) created - taskname: CLONE1_ST009 - description: DB2FIX DB2 for Db2 Cloning Tool
CKZ.SP.JCL(ST010) created - taskname: CLONE1_ST010 - description: DB2SQL for Db2 Cloning Tool
CKZ.SP.JCL(ST011) created - taskname: CLONE1_ST011 - description: DB2FIX APPL for Db2 Cloning Tool
CKZ.SP.JCL(ST012) created - taskname: CLONE1_ST012 - description: DB2SCHEMA-UPDATE for Db2 Cloning Tool
CKZ.SP.JCL(ST013) created - taskname: CLONE1_ST013 - description: DB2STOP for Db2 Cloning Tool
CKZ.SP.JCL(ST014) created - taskname: CLONE1_ST014 - description: DB2START NORMAL for Db2 Cloning Tool
```

Recloning a Db2 system

After a Db2 system has been cloned, you can reclone the system using these instructions.

1. Invoke the stored procedure with TYPE = RECLONE.

- If SP_RC equals zero, the re-cloning has completed and the target Db2 system should be running. If SP_RC is not zero, use the messages in SP_MSGS to determine the problem. It might be necessary to look at the output of the cloning jobs that have run to determine the problem.

Stored procedure example: Cloning a non-data sharing subsystem

This topic provides an example of cloning a non-data sharing subsystem, including sample parameter files and stored procedure syntax.

The following is an example of cloning a non-data sharing subsystem. The source Db2 SSID is DB2P and the target Db2 SSID is DB2T.

- Set up the product parameter file as follows:

```
CKZINI = CKZ.V32.SCKZPARM(CKZINI)
SCKZLOAD = CKZ.V32.SCKZLOAD
SCKZEXEC = CKZ.V32.SCKZPARM
SCKZDBRM = CKZ.V32.SCKZDBRM
```

- Set up the Db2 system parameter file. A sample Db2 system parameter file might appear as follows:

```
* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *
SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARM = DB2TSPEC
EXEC-SYSTEM = SYST
BDS01 = DB2T.BDS01
BDS02 = DB2T.BDS02
SYSVCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-REPORT = 3002
```

- Set up the cloning parameter file. A sample cloning parameter file might appear as follows.

```
JCL-DSN = CKZ.CLONE1.JCL
PPARM-DSN = CKZ.CLONE1.PARMS(PRODUCT)
SPARM-DSN = CKZ.CLONE1.PARMS(DB2SYS)
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
* COPY parameters
FROM-STORAGEEGROUP = DB2PSG
TO-STORAGEEGROUP = DB2TSG
USERCATALOGS = ICF.DB2P.CATALOG ICF.DB2T.CATALOG
*
* RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
RENAME-MASKS = DB2P.** DB2T.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2P DB2T
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*
SCHEMA-MASKS = DB2SCHP% DB2SCHT%
```

4. Invoke the stored procedure.

```
EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,  
:CPARMDSN, :CPARMMEM,  
:SP_RC, :SP_MSGS)
```

Where:

- CPARMDSN = CKZ.CLONE1.PARMS
- CPARMMEM = CLONE1

5. Verify the cloning.

6. Run the verified cloning.

Stored procedure example: Cloning by data set

This topic provides an example of cloning a non-data sharing subsystem by data set. The example includes sample parameter files and stored procedure syntax.

The following is an example of cloning a non-data sharing subsystem by data set. The source Db2 SSID is DB2P and the target Db2 SSID is DB2T. The data sets have a high level qualifier that is the same as the SSID.

1. Set up the product parameter file as follows:

```
CKZINI = CKZ.V32.SCKZPARAM(CKZINI)  
SCKZLOAD = CKZ.V32.SCKZLOAD  
SCKZEXEC = CKZ.V32.SCKZPARAM  
SCKZDBRM = CKZ.V32.SCKZDBRM
```

2. Set up the Db2 system parameter file. A sample Db2 system parameter file might appear as follows:

```
* * * * *  
* Source DB2 DB2P  
* * * * *  
SSID = DB2P  
SDSNLOAD = DSN.Vxxx.SDSNLOAD  
EXEC-SYSTEM = SYSA  
*  
* * * * *  
* Target DB2 DB2T  
* * * * *  
SSID = DB2T  
SDSNLOAD = DSN.Vxxx.SDSNLOAD  
SDSNEXIT = DSN.DB2T.SDSNEXIT  
SPECIAL-DSNZPARAM = DB2TSPEC  
EXEC-SYSTEM = SYST  
BDS01 = DB2T.BSDS01  
BDS02 = DB2T.BSDS02  
SYSVCAT = DB2T  
DDF-LOCATION = DB2TLOC  
DDF-LUNAME = DB2TLU  
DDF-PORT = 3001  
DDF-RESPORT = 3002
```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```
JCL-DSN = CKZ.CLONE1.JCL  
PPARM-DSN = CKZ.CLONE1.PARMS(PRODUCT)  
SPARM-DSN = CKZ.CLONE1.PARMS(DB2SYS)  
STATUS-DSN = CKZ.CLONE1.STATUS  
TASK-PREFIX = CLONE1  
USERID = CKZUSER  
PASSWORD = xxxxxxxx  
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X  
WORK-PREFIX = CKZ.CLONE1.WRK  
CLONING-TYPE = OFFLINE  
*  
* COPY-BY-DS parameters  
COPY-BY-DS = YES  
REPLACE-UNCONDITIONAL = YES  
*  
RENAME-MASKS = DB2P.BSDS0%.DATA DB2T.BSDS0%.DATA
```

```

                DB2P.LOGCOPY%.*.DATA  DB2T.LOGCOPY%.*.DATA
                DB2P.DSNDBD.**          DB2T.DSNDBD.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2P DB2T
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*
SCHEMA-MASKS = DB2SCHP% DB2SCHT%

```

4. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
:CPARMDSN, :CPARMMEM,
:SP_RC, :SP_MSGS)

```

Where:

- CPARMDSN = CKZ.CLONE1.PARMS
- CPARMMEM = CLONE1

5. Verify the cloning.

6. Run the verified cloning.

Stored procedure example: Cloning from a system-level backup

This topic provides an example of cloning a non-data sharing subsystem from a Db2 system-level backup. The example includes sample parameter files and stored procedure syntax. The Db2 system-level backup is created by using the Db2 BACKUP SYSTEM utility.

The following is an example of cloning a non-data sharing subsystem from a system-level backup. The source Db2 SSID is DB2P and the target Db2 SSID is DB2T.

1. Set up the product parameter file as follows:

```

CKZINI = CKZ.V32.SCKZPARAM(CKZINI)
SCKZLOAD = CKZ.V32.SCKZLOAD
SCKZEXEC = CKZ.V32.SCKZPARAM
SCKZDBRM = CKZ.V32.SCKZDBRM

```

2. Set up the Db2 system parameter file. A sample Db2 system parameter file might appear as follows:

```

* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *
SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARAM = DB2TSPEC
EXEC-SYSTEM = SYST
BSDS01 = DB2TLG.BSDS01
BSDS02 = DB2TLG.BSDS02
SYSVCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-RESPORT = 3002

```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```

JCL-DSN = CKZ.CLONE1.JCL
PPARM-DSN = CKZ.CLONE1.PARMS(PRODUCT)
SPARM-DSN = CKZ.CLONE1.PARMS(DB2SYS)
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1

```

```

USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
*COPY parameters
SOURCE-VOLUMES = DB2SLB
SOURCE-TOKEN = LAST
SOURCE-LOCATION = DB2PLOC
SOURCE-STORAGEGROUP = DB2PSGLG DB2PSGDB
TO-STORAGEGROUP = DB2TSGLG DB2TSGDB
USERCATALOGS = ICF.DB2P.CATALOG.LG ICF.DB2T.CATALOG.LG
                ICF.DB2P.CATALOG.DB ICF.DB2T.CATALOG.DB
*
*RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
RENAME-MASKS = DB2PLG.** DB2TLG.**
                DB2PDB.** DB2TDB.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2PDB DB2TDB
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*
SCHEMA-MASKS = DB2SCHP% DB2SCHT%

```

4. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
                                :CPARMDSN, :CPARMMEM,
                                :SP_RC, :SP_MSGS)

```

Where:

- CPARMDSN = CKZ.CLONE1.PARMS
- CPARMMEM = CLONE1

5. Verify the cloning.

6. Run the verified cloning.

Stored procedure example: Cloning from Db2 system-level backup dump tapes

This topic provides an example of cloning a non-data sharing subsystem from Db2 system-level backup dump tapes. The example includes sample parameter files and stored procedure syntax. The Db2 system-level backup dump tapes are created by using the Db2 BACKUP SYSTEM utility with the DUMP keyword.

The following is an example of cloning a non-data sharing subsystem from system-level backup dump tapes. The source Db2 SSID is DB2P and the target Db2 SSID is DB2T.

1. Set up the product parameter file as follows:

```

CKZINI = CKZ.V32.SCKZPARM(CKZINI)
SCKZLOAD = CKZ.V32.SCKZLOAD
SCKZEXEC = CKZ.V32.SCKZPARM
SCKZDBRM = CKZ.V32.SCKZDBRM

```

2. Set up the Db2 system parameter file. A sample Db2 system parameter file might appear as follows:

```

* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *

```

```

SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARM = DB2TSPEC
EXEC-SYSTEM = SYST
BDS01 = DB2TLG.BSDS01
BDS02 = DB2TLG.BSDS02
SYSVCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-RESPORT = 3002

```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```

JCL-DSN = CKZ.CLONE1.JCL
PPARM-DSN = CKZ.CLONE1.PARMS(PRODUCT)
SPARM-DSN = CKZ.CLONE1.PARMS(DB2SYS)
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
*COPY parameters
SOURCE-VOLUMES = DB2SLB
SOURCE-TOKEN = LAST
SOURCE-LOCATION = DB2PLOC
SOURCE-USE-DUMPTAPES = Y
RESTORE-DUMP-TAPE-UNIT = TAPE
RESTORE-DUMP-MAX-TAPEDRIVES = 2
RESTORE-DUMP-VARY-SCOPE = GLOBAL
RESTORE-DUMP-WAIT-TAPE-ALLOC = NO, RC(8)
SOURCE-STORAGEGROUP = DB2PSGLG DB2PSGDB
TO-STORAGEGROUP = DB2TSGLG DB2TSGDB
USERCATALOGS = ICF.DB2P.CATALOG.LG ICF.DB2T.CATALOG.LG
                ICF.DB2P.CATALOG.DB ICF.DB2T.CATALOG.DB
*
*RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
RENAME-MASKS = DB2PLG.** DB2TLG.**
                DB2PDB.** DB2TDB.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2PDB DB2TDB
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*
SCHEMA-MASKS = DB2SCHP% DB2SCHT%

```

4. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
                                :CPARMDSN, :CPARMMEM,
                                :SP_RC, :SP_MSGS)

```

Where:

- CPARMDSN = CKZ.CLONE1.PARMS
- CPARMMEM = CLONE1

5. Verify the cloning.

6. Run the verified cloning.

Stored procedure example: Cloning from Db2 system-level backup dump tapes across sysplexes

This topic provides an example of cloning a non-data sharing subsystem from Db2 system-level backup dump tapes across sysplexes. The example includes sample parameter files and stored procedure syntax.

The Db2 system-level backup dump tapes are created by using the Db2 BACKUP SYSTEM utility with the DUMP keyword. When cloning across sysplexes, the Db2 SLB dump tapes that were created on the source system must be made available for use on the target system.

A DB2GETBACKINFO job must be run on the source system to extract the required information from HSM to identify the dump tapes to be used. The DB2GETBACKINFO job creates a backinfo data set that will be used as input to the cloning job on the target system. Before running the cloning on the target system, you must transfer the backinfo data set to the target system and specify the data set name of the transferred backinfo file in the SOURCE-BACKINFO-DSN parameter in the cloning parameter file.

The following is an example of cloning a non-data sharing subsystem from system-level backup dump tapes across sysplexes. The source Db2 SSID is DB2P and the target Db2 SSID is DB2T.

1. Set up the product parameter file as follows:

```
CKZINI = CKZ.V32.SCKZPARAM(CKZINI)
SCKZLOAD = CKZ.V32.SCKZLOAD
SCKZEXEC = CKZ.V32.SCKZPARAM
SCKZDBRM = CKZ.V32.SCKZDBRM
```

2. Set up the Db2 system parameter file. A sample Db2 system parameter file might appear as follows:

```
* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *
SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARAM = DB2TSPEC
EXEC-SYSTEM = SYST
BSDS01 = DB2TLG.BSDS01
BSDS02 = DB2TLG.BSDS02
SYSCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-RESPORT = 3002
```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```
JCL-DSN = CKZ.CLONE1.JCL
PPARM-DSN = CKZ.CLONE1.PARMS(PRODUCT)
SPARM-DSN = CKZ.CLONE1.PARMS(DB2SYS)
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
*COPY parameters
SOURCE-VOLUMES = DB2SLB
SOURCE-TOKEN = LAST
SOURCE-LOCATION = DB2PLOC
SOURCE-BACKINFO-DSN = CKZ.CLONE1.BACKINFO
SOURCE-USE-DUMPTAPES = Y
RESTORE-DUMP-TAPE-UNIT = TAPE
RESTORE-DUMP-MAX-TAPEDRIVES = 2
RESTORE-DUMP-VARY-SCOPE = GLOBAL
RESTORE-DUMP-WAIT-TAPE-ALLOC = NO, RC(8)
SOURCE-STORAGEEGROUP = DB2PSGLG DB2PSGDB
TO-STORAGEEGROUP = DB2TSGLG DB2TSGDB
USERCATALOGS = ICF.DB2P.CATALOG.LG ICF.DB2T.CATALOG.LG
ICF.DB2P.CATALOG.DB ICF.DB2T.CATALOG.DB
*
*RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
```

```

RENAME-MASKS = DB2PLG.** DB2TLG.**
                DB2PDB.** DB2TDB.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2PDB DB2TDB
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*
SCHEMA-MASKS = DB2SCHP% DB2SCHT%

```

4. Set up and run the DB2GETBACKINFO job. This job must be run manually on the source system to extract the required information from HSM to identify the dump tapes to be used. The following shows a sample DB2GETBACKINFO job:

```

//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD   DISP=SHR,DSN=CKZ.V32.SCKZLOAD
//CKZINI  DD   DISP=SHR,DSN=CKZ.V32.SCKZPARM(CKZINI)
//CKZPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//ABNLIGNR DD  DUMMY
//BACKINFO DD  DSN=CKZ.CLONE1.BACKINFO,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD   DSN=CKZ.CLONE1.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD   *
DB2GETBACKINFO          -
USE-DUMPTAPES           -
BACKINFO-DDN(BACKINFO) -
WORK-DDN(HSMLIST)       -
LAST                    -
LOCATION(DB2PLOC)         -
USERCATALOGS(           -
  ICF.DB2P.CATALOG.LG   -
  ICF.DB2P.CATALOG.DB   -
)
//*

```

5. Transfer the backinfo data set created by DB2GETBACKINFO job to the target system (for example, using FTP or shared DASD).
6. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
                                :CPARMDSN, :CPARMMEM,
                                :SP_RC, :SP_MSGS)

```

Where:

- CPARMDSN = CKZ.CLONE1.PARMS
- CPARMMEM = CLONE1

7. Verify the cloning.
8. Run the verified cloning.

Chapter 11. Db2 Cloning Tool Table Space Cloning overview

Db2 Cloning Tool Table Space Cloning simplifies and automates the refresh of Db2 table spaces and index spaces. Paired with data set level fast data replication tools, data can easily be refreshed within minutes, instead of hours.

Db2 Cloning Tool Table Space Cloning's faster refresh rate alleviates troubleshooting time delays, and increases developer productivity. It is an offline utility that uses data set level fast replication tools, so it causes minimal disruption and fits into tight maintenance windows.

Db2 Cloning Tool Table Space Cloning makes it fast and easy for you to refresh Db2 test or quality assurance environments, troubleshoot production problems, and aid in development efforts because it:

- Uses high speed data set fast replication utilities, instead of traditional slow utilities
- Automates manual processes, such as object ID translation between source and target subsystems
- Uses TCP/IP to copy to subsystems not connected via CAF
- Uses an interface similar to the IBM LISTDEF facility to drastically reduce the learning curve
- Lets you select individual table spaces or an entire database
- Provides capability to exclude undesired table spaces and indexes
- Lets you select all table spaces in an RI relationship
- Easily select and migrate LOB tables and clone tables (for Db2 Version 9.1)
- Copies tables that have identity columns to another subsystem
- Beginning with Db2 Version 9.1, copies tables containing XML columns
- Includes the capability to mask column data. The changes are made based on masking rules that are enabled during the copy.
- Allows you to create the necessary jobs using ISPF interactive panels, if desired.

What does Db2 Cloning Tool Table Space Cloning do?

Cloning is the act of replicating data, making it accessible, and then using the replica in lieu of the original data for other purposes. Data set fast replication tools clone Db2 table spaces and index spaces quickly, and Db2 Cloning Tool Table Space Cloning makes the Db2 table spaces and index spaces accessible.

Fast-replicate and copy products used by Db2 Cloning Tool Table Space Cloning

Db2 Cloning Tool Table Space Cloning will make a usable clone from Db2 table spaces and index spaces created with any type of data set level fast replication utility or slow copy mechanism. Db2 Cloning Tool Table Space Cloning uses high-speed data set level fast-replicate utilities to do the copy, instead of using traditional utilities such as IBM DSN1COPY and UNLOAD/LOAD; but also works with slow copies such as DFSMSdss copy, FDR, etc.

- IBM FlashCopy
- STK SnapShot
- EMC TimeFinder/Clone Mainframe Snap Facility's data set level support

For IBM FlashCopy and STK SnapShot, Db2 Cloning Tool Table Space Cloning initiates the data set copies by executing DFSMSdss program, ADRDSSU, "under the covers." For EMC TimeFinder/Clone data set level support, Db2 Cloning Tool Table Space Cloning uses an EMC API.

IBM FlashCopy, STK SnapShot, and EMC TimeFinder/Clone allow the creation of what appears to be a copy of a data set almost instantaneously. After a copy is completed, Db2 Cloning Tool Table Space Cloning turns the resulting replicas into usable clones.

For any other cloning mechanisms, you must create the replica, using a list compiled by Db2 Cloning Tool Table Space Cloning, of data sets to copy. Db2 Cloning Tool Table Space Cloning then turns the copied data sets into usable clones.

Why you should use Db2 Cloning Tool Table Space Cloning

Db2 Cloning Tool Table Space Cloning saves time, money, and productivity in the following ways:

- Automating labor-intensive tasks increases your efficiency and reduces the potential for error
- Quicker throughput and turnaround time keeps your data available for use
- Reduces refresh costs because it is faster than traditional utilities, and eliminates time intensive manual research
- Less personnel time required to refresh Db2 table spaces and index spaces
- Allows you to manage growing environments using existing staff and within shorter time windows
- Simplifies data refresh by automating manual steps required when using traditional utilities

Can I refresh Db2 table spaces and index spaces without Db2 Cloning Tool Table Space Cloning?

Db2 table spaces and index spaces can be refreshed without using Db2 Cloning Tool Table Space Cloning by using DSN1COPY or UNLOAD/LOAD utilities, however:

- DSN1COPY requires static JCL and control parameters and does not allow for adding new Db2 extents, adding new table spaces or index spaces, or dropping existing ones. Object ID translation parameters require you to do painstaking manual research and maintenance.
- Db2 UNLOAD/LOAD can require a significant amount of time before large cloned data sets are available for use. The VSAM objects on the target side can require more space than on the source side due to FREESPACE and FREEPAGE assignments; therefore, the LOAD utility could abend and require a manual increase of space for a target table space or index space. In addition, extra time is needed to rerun the LOAD process.

Using Db2 Cloning Tool Table Space Cloning, Db2 table spaces and index spaces can be refreshed very quickly if using a fast replication utility. Db2 Cloning Tool Table Space Cloning dynamically adjusts to new or dropped table spaces and index spaces, performs automatic object ID translation, and there are no unanticipated size changes between the source and target table spaces and index spaces.

Db2 Cloning Tool Table Space Cloning features and benefits

Db2 Cloning Tool Table Space Cloning offers the following features and benefits.

- Clones Db2 table spaces and index spaces
- Is extremely fast, providing quick refresh
- Automatically invokes FlashCopy, SnapShot, or TimeFinder/Clone data set level support
- Works with both data set level fast replication utilities, and slow copy utilities
- Automates the object ID translation between the source and target Db2 table spaces and index spaces by removing the labor intensive manual research required with traditional utilities
- Includes extents automatically
- Ensures there will not be an incompatibility between the source and target table spaces and index spaces - if detected, Db2 Cloning Tool Table Space Cloning will not copy it, and provides a warning
- Uses a simple LISTDEF-like facility for ease-of-use that DBAs are already familiar with
- Supports RI relationships, LOBS, and identity columns
- Does not require a backup step
- Is not prone to copy failures due to table expansion - with Db2 Cloning Tool Table Space Cloning the target takes up the same amount of space as the source

- Copies the table spaces and index spaces outside of Db2 for faster refresh and less resource usage
- Automatically stops and starts table spaces and index spaces
- Provides integrity between the tables and indexes by stopping table spaces before index spaces
- Removes the requirement to rebuild indexes
- Optionally resets the table space time stamp
- Easier to use from start to finish than traditional utilities
- Replaces labor intensive tasks with automation to reduce errors and increase efficiency
- Reduces the need to update utility JCL in production libraries
- Simulate mode for the COPY command
- Can copy Db2 tables containing XML column data
- Optionally, when using DSS or the EMC API, Db2 Cloning Tool Table Space Cloning can tolerate enqueue failures and not stop the source spaces
- Can copy to new creator ID and/or object name

Db2 Cloning Tool Table Space Cloning terminology - Db2 objects vs. VSAM objects

Db2 Cloning Tool Table Space Cloning uses the term "objects" in two ways.

Db2 entities, such as tables, table spaces, indexes, index spaces, and applications, may be referred to as "objects", or by the specific object type, such as "table spaces" or "index spaces".

External VSAM data sets that contain table spaces and index spaces are referred to as VSAM objects.

Use of the U.S.A. EBCDIC code set

Db2 Cloning Tool uses the U.S.A. EBCDIC code set for specification and display of EBCDIC characters and for the extended ACS masking characters used for filtering.

If the code tables used by your installation are different, then you need to enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals.

- Data supplied as input to batch programs or input to ISPF panels:

For product code shipped in binary, when specifying input where the product takes special action based on specific characters, you must enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals, according to the U.S.A. EBCDIC code set.

For example, if an exclamation mark (!) is called for, and your code tables do not translate the ! character to a hexadecimal 5A, you must enter the character that your code table will translate to a 5A.

- Distributed ISPF panels:

Do not change distributed ISPF panels. Program code may reference ISPF panel attribute bytes. A panel change that affects an attribute byte may cause a program error.

- Product output:

Depictions of product output shown in the product manuals are based on the U.S.A. EBCDIC code set. Actual output may vary if your EBCDIC code tables are different.

- Extended ACS masking characters:

Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set. For more information about extended ACS masking characters, see ["Filtering pattern masks"](#) on page 26.

Chapter 12. Planning for copying Db2 table spaces by data set

Before attempting to actually use Db2 Cloning Tool Table Space Cloning, some planning and decision-making should take place. This topic discusses those things that need to be considered.

Considerations for in-progress read/write activity or Db2 utilities

Db2 utility activity or other read/write activity can affect Db2 Cloning Tool Table Space Cloning processes.

Db2 table spaces and index spaces being copied should not have a Db2 utility in progress. In addition, there should not be any read/write activity against source or target table spaces or index spaces to be copied (unless the FUZZY copy option is chosen).

Db2 Cloning Tool Table Space Cloning issues a Db2 STOP command to the specific source and target Db2 table spaces and index spaces being copied. The Db2 STOP command waits with a pending status (STOPP) until the table spaces and index spaces are no longer in use. There is no way to force activity to stop. Only a successful STOP command will deallocate the associated VSAM data set from DBM1. If the table spaces or index spaces are in use, Db2 Cloning Tool Table Space Cloning waits a user-specified number of seconds for the STOP command to complete successfully.

Considerations for the Db2 Cloning Tool Table Space Cloning cloning process

The following rules must be taken into consideration when using Db2 Cloning Tool Table Space Cloning.

Subsystem copy rules

Table spaces and indexes spaces can be copied to the same Db2 subsystem or a different Db2 subsystem.

Db2 Cloning Tool Table Space Cloning can copy table spaces and index spaces between an earlier version of Db2 to a later version of Db2, allowing you to easily port spaces for a Db2 version upgrade.

The Db2 Cloning Tool Table Space Cloning OBJECT-TRANSLATE keyword allows the copying of table spaces and index spaces to the same subsystem. Once the source table spaces and index spaces have been determined, the same database and space name are used for the target data sets unless object translation is specified. If object translation is specified, Db2 Cloning Tool Table Space Cloning attempts to use the supplied OBJXLAT commands to map the existing database and space names into new target data sets.

When processing object translation, Db2 Cloning Tool Table Space Cloning does not verify that the resulting target index will correctly point to the resulting target table. However, Db2 Cloning Tool Table Space Cloning does verify that the table spaces and index spaces are compatible from the source to the target. Object translation should be carefully specified to ensure proper results.

Renaming table spaces and index spaces

Table spaces and index spaces cannot be renamed because Db2 does not support it.

A rename of a Db2 table space or index space can only be accomplished by using the Db2 DROP and CREATE commands. A Db2 DROP deletes both the VSAM data set and the Db2 catalog information concerning the Db2 table space or index space. However, a source table space or index space can be copied to an existing target table space or index space with a different name.

Implicitly created objects

Db2 Cloning Tool Table Space Cloning can automatically match the names of source and target implicitly created objects.

The following implicitly created objects are supported:

- Table space
- LOB table space
- Auxiliary table
- XML table space
- XML table
- Index and index space

If source objects (XML, LOB, or base table spaces and their indexes) were created explicitly and target objects (XML, LOB, or base table spaces and their indexes) were created implicitly, AUTO-TABLESPACE-TRANSLATE and AUTO-INDEXSPACE-TRANSLATE parameters should be specified so that source and target objects can be matched (assuming that OBJECT-TRANSLATE was not specified for these objects). If source objects were created implicitly, the AUTO-TABLESPACE-TRANSLATE and AUTO-INDEXSPACE-TRANSLATE parameters are not required.

The following logic is used in matching the source and target names for implicit source objects:

1. If the matching target object is found, Db2 Cloning Tool Table Space Cloning uses the target names.
2. If the matching target object is not found, and OBJECT-TRANSLATE is specified for the source object, Db2 Cloning Tool Table Space Cloning uses the target names specified in OBJECT-TRANSLATE.
3. If the matching target object is not found, and OBJECT-TRANSLATE is not specified for the source object, the result of copying depends on the actual name of the target object and other Db2 Cloning Tool Table Space Cloning settings. Refer to the topic [“Target analysis”](#) on page 209 for more information about how Db2 Cloning Tool Table Space Cloning processing confirms that a compatible target table space or index space exists on the target Db2 subsystem.

Limitations

To enable automatic matching for implicit indexes, you must include the source table spaces or tables for those indexes in the LISTDEF.

If only the target objects are implicit (the source objects are explicit), OBJECT-TRANSLATE must be specified for those objects.

Identity columns

Db2 Cloning Tool Table Space Cloning can update the sequence numbers for identity columns in the Db2 catalog.

Db2 Cloning Tool Table Space Cloning issues ALTER TABLE SQL to correct identity column values. If you are using DDL generation (DDL-ENABLE set to Y), and either PROCESS-TYPE(Y) or PROCESS-TYPE(G) are specified, then the ALTER TABLE SQL statements for identity columns are included in the generated DDL. This DDL is either run or saved to a data set, depending on the value of the PROCESS-TYPE keyword. If DDL-ENABLE(Y) and PROCESS-TYPE(N) are specified, or DDL-ENABLE(N) and PROCESS-TYPE(N) are specified, then the ALTER TABLE SQL statements for identity columns are generated and run.

Simulation mode affects whether identity columns are processed. With SIM(A), the following logic is followed:

- If DDL-ENABLE(N) and SIM(A), ALTER TABLE SQL syntax is not generated or run for identity columns.
- If DDL-ENABLE(Y) and SIM(A), the PROCESS-TYPE setting defines how actions with generated DDL and ALTER TABLE SQL for identity columns are processed, as follows:

- If PROCESS-TYPE(N), PROCESS-TYPE(G), or PROCESS-TYPE(A), neither DDL nor ALTER TABLE for identity columns is run.
- If PROCESS-TYPE(X), DDL that is provided is run, but ALTER TABLE SQL for identity columns is not generated or run.
- If PROCESS-TYPE(Y), both DDL and ALTER TABLE SQL for identity columns are generated and run.

LOBs, BLOBs, CLOBs, and DBCLOBs

When copying a large object (LOB) table, both the table space containing the LOB column and the table space containing the auxiliary table with the LOB data must be copied at the same time using the LISTDEF ALL command.

The target object or objects may not be accessible if BASE or LOB LISTDEF keywords are specified.

XML considerations

This topic describes the considerations for cloning table spaces that contain XML column data.

Cloning tables containing XML column data

If the source and the target Db2 subsystems are not on the same LPAR, XML processing requires that you use the source TCP/IP server job or started task.

XML string processing

Db2 Cloning Tool Table Space Cloning must add string IDs to the target SYSIBM.SYSXMLSTRINGS catalog table when the source and target subsystems are different and the XML column(s) contain string IDs in the source table space that are not in the target table space, or that are mapped to different string IDs. Db2 Cloning Tool Table Space Cloning adds strings into the target catalog table SYSIBM.SYSXMLSTRINGS. The strings IDs are translated from the source to the target value within the data pages of XML columns.

Requirements when the target is a member of a data sharing group

When Db2 Cloning Tool Table Space Cloning connects to the target SSID in the target job, it updates the sequence number used for the document ID column DB2_GENERATED_DOCID_FOR_XML.

However, when the target is a member of a data sharing group, Db2 Cloning Tool Table Space Cloning updates only the sequence number for the target, which Db2 Cloning Tool Table Space Cloning is connected to. For other members of the data sharing group, you must specify a keyword so that Db2 Cloning Tool Table Space Cloning generates the jobs to update the sequence number for other members of the data sharing group. Each of these jobs must be run on the appropriate LPAR to update the sequence number that is used for the DB2_GENERATED_DOCID_FOR_XML column.

This process is required due to how sequence number columns are managed within a data sharing group. Each member of a group allocates blocks of sequence numbers that are used as needed by that particular member. If a sequence number must be increased, and the update job is run on a member that has already allocated a block of values that exceed the target value, another member may not necessarily be set to a high enough value. The Db2 Cloning Tool Table Space Cloning jobs that are generated for each member must be run to reset the sequence numbers and prevent this problem.

Therefore, when the target Db2 subsystem is a member of a data sharing group, you must specify the SET keyword UPDATE-DOCID-JCL-DSN in the source job. When the job is submitted, Db2 Cloning Tool Table Space Cloning generates SYNCDOCID commands in the SYNCDB2 file for each data sharing group member that needs to have the DOCID updated.

Other XML considerations

When copying a table with an XML column, both the table space containing the XML column and the table space containing the XML data should be copied at the same time using the LISTDEF ALL command. If you use only BASE or XML LISTDEF keywords, the target object or objects may not be accessible.

TCP/IP considerations

If the target subsystem is on a different LPAR than the source subsystem, Db2 Cloning Tool Table Space Cloning requires that a TCP/IP connection be available from the source z/OS subsystem to a Db2 Cloning Tool Table Space Cloning TCP/IP server on the target z/OS system, and from the target z/OS system to a Db2 Cloning Tool Table Space Cloning source TCP/IP server on the source z/OS system.

A TCP/IP server is used to access the target catalog and to issue Db2 commands. Communication with the target is how Db2 Cloning Tool Table Space Cloning can verify compatibility between source and target Db2 table spaces and index spaces. This communication also allows it to acquire and translate the internal object identifiers between the source and the target Db2. A source TCP/IP server job is used for XML and log apply processing during cross-LPAR cloning. For TCP/IP communication, Db2 Cloning Tool Table Space Cloning provides server jobs CKZTCPS and CKZTCPSS. These jobs are available in the product JCL library.

Note: Use SET REMOTE-CONNECT-TYPE to specify CAF or TCP/IP.

APF authorization

The Db2 Cloning Tool Table Space Cloning LOAD library must be APF authorized.

This is included as a configuration step in [“APF authorizing load libraries” on page 30](#).

Ensure RACF and ACF authorities

Before cloning, you should ensure that all proper RACF and ACF permissions and privileges are in place.

Refer to [“Verify that your environment meets security requirements” on page 22](#) for information about required permissions and privileges for Db2 Cloning Tool Table Space Cloning.

Missing target table spaces and index spaces

Db2 Cloning Tool Table Space Cloning uses a single target high level qualifier for all VSAM objects associated with Db2 table spaces and index spaces missing from the target Db2 catalog. It will not be able to acquire the VCAT (high level qualifier) for the missing table spaces and index spaces from the Db2 catalog. Db2 Cloning Tool Table Space Cloning will use the DEFVCAT (default VCAT) parameter supplied in the COPY command for all such VSAM data sets.

Note: If a Db2 table space or index space is created on the target, after the VSAM object is copied from the source, there can be a problem: When Db2 is instructed to create the target table space or index space, it will place the target internal object identifiers inside the existing VSAM object. That immediately makes any existing data inside the VSAM object permanently inaccessible. The solution is to make the name of target VSAM object unrecognizable by Db2. Allow Db2 to create a new VSAM object, then delete that object and rename the target VSAM object to make it recognizable to Db2.

Considerations for target objects created using DEFINE NO

If a Db2 table space or index space is created on the target with DEFINE NO, the data set for the object will not exist until a row is inserted into the object. The target Db2 subsystem must be forced to create target data sets for the “DEFINE NO” objects.

If the target Db2 subsystem does not create the target data sets, Db2 will never access the data sets cloned from the source Db2. When the target data set does not exist, Db2 Cloning Tool Table Space Cloning will copy the object as if the target object does not exist. In this case, you may force the target Db2 subsystem to create the target data sets either before the source and target jobs execute or after they execute.

If the target data set for the “DEFINE NO” object is created before the source and target jobs execute:

No special actions are required; cloning works as it works with DEFINE YES objects.

1. Execute the source job.
2. After the source job completes successfully, execute the target job.

If the target data set for the “DEFINE NO” object will be created after the source and target jobs execute:

The cloning process results in the following behavior, regardless of the parameter value of COPY-IJ-TO-NONEXISTENT-TARGET. Take the following steps:

1. Submit the source job. The source job will target output data sets with the fifth node qualifier of *.F0001.* (i.e. HLQ.DSNDB*.DBNAME.TSNAME.F0001.Annn) and the target VCAT taken from the target catalog. This is done so that when the target Db2 attempts to create an *.I0001.* data set, it will not find the one copied from the source already in existence. If Db2 found an *.I0001.* data set, it would initialize the existing data set and all data would become inaccessible.
2. Submit the target job. The target job will update object IDs in the *.F0001.* data set(s) to match those in the target catalog. It will also automatically start the target object(s). To assist you in renaming the data sets, IDCAMS parameters are written to an optional IDCAMS-DDN file. These parameters may be used to rename the *.F0001.* data set(s) to *.I0001.* data set(s).
3. You must then complete the following steps:
 - a. Insert a row into the target table. If the table is partitioned, then a row must be inserted into the desired partition(s).
 - b. Stop the target object(s). This prevents enqueue conflicts with the IDCAMS job over the *.I0001.* data sets.
 - c. Submit a job using the IDCAMS parameters to rename the *.F0001.* data set(s) to *.I0001.* data set(s). This job will delete the *.I0001.* data sets created by Db2 and then rename the *.F0001.* data sets to *.I0001.* data sets.
 - d. Start the target object(s).

Note: If the source job is submitted with the DATA-MOVER(PGM(NONE)) keyword, the -STOP parameters for all target objects will be written to the optional STOP-TARGET-DDN file. These parameters may then be copied and edited to create -START parameters for the target objects.

The easiest way to copy using Db2 Cloning Tool Table Space Cloning is to define the target object with a minimum size. Then when Db2 Cloning Tool Table Space Cloning copies the source data, the old data set is deleted and a new larger data set is allocated. No additional processing would be required.

If it is not known for certain whether any target objects were created with DEFINE NO and do not have a data set created by Db2, submit a Db2 Cloning Tool Table Space Cloning source job with COPY-IF-NO-DB2-TARGET-OBJECTS(N) and PGM(NONE). All target table space and index spaces without a data set will be listed in the output with a warning message.

Considerations for DFSMS pervasive encryption

When cloning table spaces in an environment where DFSMS pervasive encryption is in use, review the following items to ensure a successful cloning.

Copying objects that are encrypted with DFSMS pervasive encryption

Db2 Cloning Tool supports Db2 objects on Db2 V11 and Db2 V12 that are encrypted with the DFSMS pervasive encryption. Supported cloning scenarios and environmental requirements depend on the value of DATA-MOVER PGM keyword, as follows:

- For ADRDSSU: For each source and target data set pair, both data sets should be either encrypted or non-encrypted. For cross-LPAR cloning scenarios, the DFSMS key label used for source data set encryption should be available on the target LPAR and should refer to the same encryption key as on the source LPAR.
- For EMCAPI: DFSMS pervasive encryption is not supported.
- For SRCIMCPY and SRCVSCPY: Source and target data sets may have any encryption status. For cross-LPAR cloning scenarios, the DFSMS key label that is used for the target data set encryption should be available on the source LPAR and should refer to the same encryption key as on the target LPAR.
- For NONE: In each source and target data set pair, both data sets should be either encrypted or non-encrypted. For cross-LPAR cloning scenarios, the DFSMS key label used for the source data set encryption should be available on the target LPAR and should refer to the same encryption key.

As a part of the table space cloning source job, Db2 Cloning Tool verifies that each source and target data set pair satisfies above restrictions. If a source and target data set pair does not satisfy any of those restrictions, Db2 Cloning Tool treats that as the encryption mismatch.

When the encryption mismatch is detected, the behavior of Db2 Cloning Tool is controlled by the ENCRYPTION-MISMATCH-RC, ALLOW-COPY-ON-MISMATCH, and EXCLUDE-MISMATCH-PROCESSING keywords of the COPY command. If EXCLUDE-MISMATCH-PROCESSING(Y) is used, Db2 Cloning Tool does not perform any encryption-related checks. If ALLOW-COPY-ON-MISMATCH(Y) is used, objects are not be excluded from the copy, even if an encryption mismatch is detected. ENCRYPTION-MISMATCH-RC controls the severity of all encryption-related mismatches. Setting this keyword to 0, 4, or 8 causes Db2 Cloning Tool to ignore encryption mismatches or treat them as a warnings or errors. However, there is one encryption mismatch that cannot be ignored with ENCRYPTION-MISMATCH-RC: for ADRDSSU, the encryption status of the source and target data sets must be the same. Ignoring this mismatch would cause the copy to fail or would result in an unusable target data set.

If cloning parameters allow the use of UNLOAD/LOAD or REBUILD INDEX utilities, Db2 Cloning Tool attempts to use them to resolve encryption-related mismatches.

When Db2 Cloning Tool needs to exclude a source and target data set pair from the copy due to an encryption mismatch, all other data sets for this object and its related objects are excluded from the copy.

Using DFSMS pervasive encryption to encrypt Db2 Cloning Tool work data sets

Depending on cloning parameters, Db2 Cloning Tool might dynamically create additional data sets during the cloning process. These data sets include log apply sort files, work files, minilog data sets, UNLOAD/LOAD utility SYSPUNCH, SYSREC, SORTOUT, and SYSUT data sets, and real-time statistics (RTS) data sets.

Important: These data sets might contain sensitive information (such as the contents of Db2 logs, object data records, or RTS for cloned objects). If DFSMS encryption is required for these data sets, you must provide the corresponding allocation parameters.

To allocate the log apply work file, sort file or minilog data sets as encrypted, specify the SMS data class with the defined encryption key label in the LOG-APPLY keywords WORKFILE-DATACLAS, SORTFILE-DATACLAS, and MINILOG-DATACLAS keywords. For minilog data sets, you also can use a RACF data set profile and specify the corresponding value in the MINILOG-HLQ keyword.

To allocate RTS data sets as encrypted, specify the SMS data class with the defined encryption key label in the RTS-COPY keyword RTSFILE-DATACLAS, or use a RACF data set profile and specify the corresponding value in the RTSFILE-DATA-SET-HLQ keyword.

To allocate UNLOAD/LOAD data sets as encrypted, specify the SMS data class with the defined encryption key label, or use a RACF data set profile and specify the corresponding HLQ in the TEMPLATE-SORTOUT-DDN, TEMPLATE-SYSPUNCH-DDN, TEMPLATE-SYSREC-DDN, and TEMPLATE-SYSUT-DDN parameters of the UNLOAD-LOAD keyword.

Additional authorization requirements

The Db2 Cloning Tool user ID must be permitted to use any key labels that are used to protect data sets that are involved in the cloning process. This includes any data set that is referenced by a DD statement in JCL, and data sets from the following list:

For the source job:

- Source Db2 data sets for cloned objects.
- FlashCopy consistent image copy data sets, if USE-LAST-CONSISTENT-FLASHCOPY(Y) is used.
- Image copy data sets, if the SRCIMCPY DATA-MOVER PGM is used.
- Target Db2 data sets for cloned objects, if CAF is used for the target Db2 connection.
- UNLOAD/LOAD SYSPUNCH, SYSREC, SORTOUT, and SYSUT data sets, if UNLOAD/LOAD is used.
- Data sets under the HLQ that is specified in RTS-DATA-SET-HLQ, if RTS copy is used.
- Target Db2 data sets for cloned objects, source Db2 log data sets, data sets under MINILOG-HLQ, and work file and sort file data sets, if SRCIMCPY or SRCVSCPY DATA-MOVER programs are used.

For the target job:

- Target data sets, except when SRCIMCPY or SRCVSCPY DATA-MOVER programs are used.
- UNLOAD/LOAD SYSPUNCH and SYSREC data sets, if UNLOAD/LOAD is used.
- Source Db2 log and BSDS data sets, if the source job used CAF for the target Db2 connection and target log apply is used
- Data sets under MINILOG-HLQ, if target log apply is used.
- Data sets under the HLQ that is specified in RTS-DATA-SET-HLQ, if RTS copy is used.
- A data set that is referenced by the UPDATE-DOCID-JCL-DSN keyword.

For the target server job, authorization is required for the target Db2 data sets for cloned objects.

For the source server job, if target log apply is used authorization is required for the source Db2 log and BSDS data sets, data sets under MINILOG-HLQ, and work file and sort file data sets.

Non-SMS managed volumes

If Db2 Cloning Tool Table Space Cloning is controlling the copy process and the target volumes are non-SMS managed, the volsers needed for original data set(s) and the target data set(s) allocation may be passed to ADDRSSU via the HLQDDDF command as defined in the Db2 Cloning Tool Table Space Cloning source job, as shown in this topic.

The HLQDDDF DIRECTION(IN) parameter passes the volsers for input volumes using the DD names from the source job JCL:

```
MYINDD DD DISP=SHR,UNIT=3390,VOL=SER=(JM505F,JM515F)
MYINDD2 DD DISP=SHR,UNIT=3390,VOL=SER=(JM525F,JM535F)
```

Similarly, DIRECTION(OUT) passes volsers for output volumes using the DD names from the source job JCL:

```
MYOUTDD DD DISP=OLD,UNIT=SYSDA,VOL=SER=(JM505A,JM515A)
MYOUTDD2 DD DISP=OLD,UNIT=3390,VOL=SER=(JM525A,JM535A)
```

The command HLQDDDF may occur multiple times in the CKZIN DD, and keyword DDNAME can pass multiple DDnames:

```
CKZIN DD *
HLQDDDF HLQNAME(FDRABR1) DIRECTION(IN) DDNAME(MYINDD,MYINDD2)
HLQDDDF HLQNAME(FDRABR2) DIRECTION(OUT) DDNAME(MYOUTDD,MYOUTDD2)
```

Space processing order

Db2 Cloning Tool Table Space Cloning always processes table spaces first, followed by index spaces.

How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces

Db2 Cloning Tool Table Space Cloning starts and stops source and target table spaces and index spaces depending on certain parameter settings.

Source table spaces and index spaces

For source table spaces and index spaces, the Db2 Cloning Tool Table Space Cloning COPY command keywords FUZZY-COPY and AUTO-START-SOURCE-SPACE control whether source table spaces and index spaces are stopped and started before and after the copy.

Note: Db2 stops and starts are done only if the COPY command keyword DATA-MOVER is set to PGM(ADDRDSSU), PGM(EMCAPI), PGM(SRCIMCPY), or PGM(SRCVSCPY). DATA-MOVER PGM(NONE) does not start or stop any Db2 objects. However, for stopped source table spaces and index spaces that require processing with UNLOAD, if AUTO-START-SOURCE-SPACE(Y), the spaces are started before utility execution, regardless of the DATA-MOVER type or FUZZY-COPY value.

- If COPY command keyword FUZZY-COPY(Y) is in effect, no stops or starts will be applied to the source table spaces and index spaces. Table spaces and index spaces are copied in their current state.
- If COPY command keyword FUZZY-COPY(N) has been specified, before any copies begin, all source table spaces and index spaces are stopped. After all copies are completed, all source spaces are started according to the AUTO-START-SOURCE-SPACE keyword. COPY command keyword SIMULATE(A) or SIMULATE(N) will both result in source table spaces and index spaces being stopped and started when FUZZY-COPY(N) is in effect.

If an error occurs when attempting to stop a source space, Db2 Cloning Tool Table Space Cloning tries to restart all the source spaces that were stopped to their initial status.

Target table spaces and index spaces

For target table spaces and index spaces, the Db2 Cloning Tool Table Space Cloning COPY command keywords or INI tokens AUTO-STOP-TARGET-SPACE and AUTO-START-TARGET-SPACE, and the COPY command keyword SIMULATE (or SIM), control stopping and starting of target table spaces and index spaces.

- COPY command keyword or INI token AUTO-STOP-TARGET-SPACE: If set to Y, the source job stops the target table spaces and index spaces.
 - If SIM(A) is in effect, the source job restarts the target table spaces and index spaces, as the target job will not run until the table spaces and index spaces are copied with SIM(N).
 - If SIM(N), the table spaces and index spaces remain stopped.

Note: If AUTO-STOP-TARGET-SPACE(Y) is specified, then the target table spaces and index spaces will be stopped even if FUZZY-COPY(Y) is in effect.

- COPY command keyword or INI token AUTO-START-TARGET-SPACE: This keyword or INI token determines whether the Db2 Cloning Tool Table Space Cloning target job starts the spaces after SYNCDB2 processing is completed on the target Db2 subsystem. If AUTO-START-TARGET-SPACE(Y), each space is started as soon as its SYNCDB2 processing is completed. Stopped target table spaces and index spaces that require processing with the LOAD or REBUILD INDEX utilities are started before utility execution. Stopped target index spaces that require processing with the REBUILD INDEX utility are started even if SIM(A).

If an error occurs while attempting to stop a target space:

- If SIM(A) is in effect, Db2 Cloning Tool Table Space Cloning tries to restart all the target spaces that were started when Db2 Cloning Tool Table Space Cloning was invoked.

- If SIM(N) is in effect, the target spaces will not be restored to their pre-Db2 Cloning Tool Table Space Cloning status. Db2 Cloning Tool Table Space Cloning assumes the user will stop the target table spaces and index spaces in STOPP (stop pending) outside of Db2 Cloning Tool Table Space Cloning and then rerun the source job.

When target spaces are restarted due to a STOPP error, Db2 Cloning Tool Table Space Cloning remembers the status of each table space and index space [RO (read only) or RW (read-write)] and issues the correct START command. Note that in the target job all target spaces are started RW.

If table spaces and index spaces do not exist on the target, no commands are issued.

Commands needed to stop and start table spaces and index spaces can be obtained by running the Db2 Cloning Tool Table Space Cloning source job with PGM(NONE). This can be useful if manual commands must be entered. Specify STOP-TARGET-DDN, STOP-SOURCE-DDN and START-SOURCE-DDN in the COPY command.

Message output

If warning or error messages are being output and are difficult to find in the CKZPRINT data set, you can route the warning and error messages to a separate DD.

Add the CKZERROR DD to the source, target, TCP/IP server, and/or source TCP/IP server jobs. When this DD is present, all warning and error messages will be output to this DD, as well as to CKZPRINT. For example:

```
//CKZERROR DD SYSOUT=*
```

Note: If the CKZERROR DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.

Object mismatch processing in Db2 Cloning Tool Table Space Cloning

When cloning objects from the source subsystem to a target subsystem, and the target objects already exist on the target subsystem, certain object attributes must match in order for the source objects to be successfully cloned to the existing target objects. Db2 Cloning Tool Table Space Cloning performs attribute checking as part of the cloning process. All objects with mismatched attributes are identified and reported, and Db2 Cloning Tool Table Space Cloning also can automatically process some mismatches.

There are several options for resolving object mismatches:

- You can specify the RECREATE-OBJECTS(M) keyword to drop and re-create target objects that have mismatches with the source objects.
- If keeping the existing target objects is not important, you can specify the RECREATE-OBJECTS(A) keyword. Db2 Cloning Tool Table Space Cloning will create DDL to drop and re-create all target objects.
- If other methods fail, you can use UNLOAD/LOAD to resolve mismatches.

If Db2 Cloning Tool Table Space Cloning detects a mismatch, a warning message with a return code of 4 is generated in the output. You can override the return codes that are assigned to specific mismatch types by using the OBJECT-MISMATCH-RETURN-CODE keyword in the COPY command. The return code and the warning message that is issued can be 0 (informational message), 4 (warning message), or 8 (error message).

Note: Using EXCLUDE-MISMATCH-PROCESSING(Y) is not recommended. You should ensure that all mismatches have been identified and corrected prior to running the source job.

The following topics provide details about object attribute checking and logic, as well as describe how you can set up automatic processing of mismatches.

Object attributes that must be identical between source and target

The attributes and contents of the source Db2 table space and index space data sets must not conflict with the attributes of the target Db2 table space and index space data sets.

For example, they must have the same object types, buffer pool sizes, DSSIZE, and use the same code page (CCSID).

The table columns and column attributes of the source and target tables must be identical. Other object attributes that must be identical are listed in the following table:

| Object type | Attributes that must be identical for source and target object |
|-------------------------|---|
| Database | None |
| Table space | CCSID, CLONE, COMPRESS, DSSIZE, NTABLES, ORGANIZATIONTYPE, PAGENUM, PARTITIONS, PGSIZE, SEGSIZE, TYPE |
| Partitioned table space | COMPRESS, DSSIZE, FORMAT, LIMITKEY |
| Table | COLCOUNT, EDPROC, HASHKEYCOLUMNS |
| Column | CACHE, COLNO, COLTYPE, CYCLE, DEFAULT, FLDPROC, HASHKEY_COLSEQ, INCREMENT, LENGTH, MAXVALUE, MINVALUE, SCALE, SEQTYPE, SOURCETYPEID, START |
| Index | COLCOUNT, COLNO, COLSEQ, CLUSTERING, COMPRESS, DSSIZE, HASH, INDEXTYPE, ORDERING, PAGENUM, PIECESIZE, PGSIZE, SPARSE, UNIQUE_COUNT, UNIQUERULE; also the number of partitions in the index must be identical. |
| Partitioning index | DSSIZE, LIMITKEY |

In general, differences that affect the layout of the data or the data set prevent successful synchronization between source and target table spaces and index spaces. Attributes that affect only how Db2 processes the data, such as LOCKRULE, can be different.

How Db2 Cloning Tool Table Space Cloning checks object compatibility

During object compatibility checking, Db2 Cloning Tool Table Space Cloning operates on groups of objects. How Db2 Cloning Tool Table Space Cloning determines if a group of objects is compatible is based on several factors.

A group of objects consists of a base table space with its tables, the indexes on these tables, XML and LOB spaces for this table space and their indexes, and all RI-related objects with their indexes and XML and LOB spaces. If XML and LOB spaces are cloned without the base table space, they are each considered as a separate group (a group includes the XML AUX indexes). Indexes that are cloned without their table spaces are each considered as a separate group.

If a table space from a group must be excluded from the cloning process, the entire group is excluded from cloning process. If an index or index partition must be excluded from the cloning process, only the index or its partition is excluded from cloning process; the remainder of the group is still processed. Groups are used to keep set of cloned objects consistent.

Db2 Cloning Tool Table Space Cloning performs the following steps to check object compatibility.

Note: Steps 1-5, 7, and 10 are not performed if object mismatch processing is disabled (due to EXCLUDE-MISMATCH-PROCESSING(Y)).

1. Catalog field values of mapped objects are compared. All mismatches are printed for each table space, its partitions, tables, and their columns, for each index and its partitions. During this step, objects with mismatches that can be processed by UNLOAD/LOAD and indexes with mismatches are not excluded for separate processing.

2. If RECREATE-OBJECTS(M) is specified, objects that are marked for re-creation are checked for re-creation compatibility. If an object is not compatible with re-creation, Db2 Cloning Tool Table Space Cloning attempts to use other methods of resolving mismatches, such as UNLOAD/LOAD or REBUILD. Refer to [“Re-creating existing target objects”](#) on page 196 for more information about re-creating target objects.
3. If UNLOAD/LOAD is enabled, objects that are marked for UNLOAD/LOAD are checked for UNLOAD/LOAD compatibility. Note that if DDL generation and execution is enabled, this step is run after completion of DDL generation, DDL execution, and the second catalog reading, because some mismatches might be fixed during DDL generation. If an object with a mismatch that can be processed by UNLOAD/LOAD has RI relationships or a column with inline LOB length greater than 32Kb, then the object is not compatible with UNLOAD/LOAD. If an object does not meet these restrictions, then Db2 Cloning Tool Table Space Cloning checks if this object and all other objects in its group have mismatches that are not compatible with UNLOAD/LOAD. Those include object mismatch return codes of 8, or object mismatch return code of 4 and ALLOW-COPY-ON-MISMATCH is set to N. Index and index part mismatches with object mismatch return codes less than 8 are not considered as problems in this case, because LOAD will REBUILD those objects. A list of mismatches compatible with UNLOAD/LOAD can be found in the [UNLOAD-LOAD](#) keyword description in the [“COPY command and keyword definitions”](#) on page 516 topic.
4. All incompatible objects along with their groups are excluded from the cloning process.
5. Version mismatches are checked for the remainder of the compatible objects (except ones that will be processed by UNLOAD/LOAD). When PROCESS-TYPE(G) or PROCESS-TYPE(Y) is used for DDL generation, and a source object table exists with a non-zero version and target REPAIR utility execution is disabled, the object will be excluded as incompatible and will not be processed for DDL generation, because cloning will make target object inaccessible.
6. DDL generation and execution takes place if needed.
7. If target objects were changed or created during DDL generation, they are rechecked for object mismatches (see steps 2-6).
8. If a source object has no target object and COPY-IF-NO-DB2-TARGET-OBJECTS is set to N, the entire group for this object is excluded from cloning.
9. If a problem occurs during target data set preparation or data set existence check, and object cannot be copied, the entire group for this object is excluded from cloning process.
10. Extension mismatches are checked.
11. Indexes that should be rebuilt are excluded from cloning process and will be rebuilt in the target job.

For each object that is excluded from the cloning due to object mismatches, Db2 catalog processing errors, target data set processing errors, non-existent target objects, encryption mismatches, or UNLOAD/LOAD incompatibility, warning messages CKZ54303W, CKZ54307W, CKZ54308W, CKZ54309W, CKZ54311W, and CKZ54313W provide a group number and the reason for exclusion. When the reason for exclusion of a group cannot fit on a single line, message CKZ54317W is issued after the list of excluded objects; it includes a group number and the reason for group exclusion.

The rest of the cloning process does not use groups.

Note: If PROCESS-TYPE(G) is specified, all compatible objects are still processed via DDL generation even if mismatches with return code of 8 are found. After the DDL generation process has completed, Db2 Cloning Tool Table Space Cloning finishes with an error. In other cases, Db2 Cloning Tool Table Space Cloning finishes with an error after all mismatches are checked.

Strategies for resolving object mismatches

Resolve mismatches by re-creating target objects using DDL, or use the UNLOAD-LOAD keyword in the COPY command.

Resolving mismatches by re-creating target objects

You can resolve mismatches by specifying the RECREATE-OBJECTS(M), IGNORE-DROP-OBJECT-DOES-NOT-EXIST(Y), RECREATE-TABLE-RESTRICTED-ON-DROP(Y), and IGNORE-CREATE-OBJECT-EXISTS(Y) keywords in the COPY command PROCESS-DDL parameter.

Db2 Cloning Tool resolves the object mismatches by dropping the target objects and re-creating them with the same attributes as the corresponding source objects.

To re-create all target objects regardless of mismatch status, specify RECREATE-OBJECTS(A). This will generate DROP and CREATE statements for all target objects.

Re-creating objects can resolve most mismatches. For information about re-creating objects, see the topic [Re-creating objects with mismatches](#).

Using UNLOAD/LOAD to resolve object mismatches

If mismatches cannot be resolved using other methods, UNLOAD/LOAD can be used. However, UNLOAD/LOAD might be less efficient than using the other methods.

UNLOAD/LOAD can be used to resolve the following mismatches:

- TS_CLONE
- TS_COMPRESS
- TS_COMPRESS_TYPE
- TS_DSSIZE
- TS_PAGENUM
- TS_PARTITIONS
- TS_PGFSIZE
- TS_SEGSIZE
- TP_COMPRESS
- TP_DSSIZE
- TP_LIMITKEY
- CL_LENGTH (only if the length of the source column is less than the length of the target column)

There are restrictions to using UNLOAD/LOAD. See the [UNLOAD-LOAD keyword description in the “COPY command and keyword definitions” on page 516](#) topic for detailed information about UNLOAD/LOAD.

Tips for specific mismatches

Some types of mismatches might require special attention.

Mismatch: Source partition-by-growth table space has more partitions than the corresponding target

If your installation of Db2 Cloning Tool V3.2 does not have PTF UI54978 applied, it is recommended that you always specify EXTEND-TARGET-PBG-TABLESPACE(Y). After PTF UI54978 is applied, the EXTEND-TARGET-PBG-TABLESPACE parameter is deprecated, and Db2 Cloning Tool Table Space Cloning always automatically extends target partition-by-growth table spaces.

In addition, PTF UI61255 includes fixes for the processing of mismatches for XML partition-by-growth table spaces. If you plan to clone XML spaces, you should apply PTF UI61255.

Mismatches: Record format (BRF vs. RRF), actual page format vs. catalog column RBA_FORMAT, HASHDATAPAGES, table versioning mismatches

To automatically resolve these mismatches, it is recommended that you always specify TARGET-JOB-REPAIR-EXECUTE(Y) and OBJECT-MISMATCH-RETURN-CODE(TP_FORMAT, 0, TS_HASHSPACE, 0).

For table versioning mismatches, you should apply Db2 Cloning Tool V3.2 PTF UI54552 so that table versioning mismatch processing can be successfully completed. Additional information related to Db2 Cloning Tool table versioning mismatch processing can be found here:

<http://www-01.ibm.com/support/docview.wss?uid=swg22015834>

<http://www-01.ibm.com/support/docview.wss?uid=swg22015681>

Mismatch: Column cache

This mismatch can be ignored. To automatically resolve this mismatch between source and target columns, it is recommended to always specify OBJECT-MISMATCH-RETURN-CODE(CL_CACHE, 0).

Mismatch: Identity column values

Db2 Cloning Tool Table Space Cloning always generates ALTER TABLE RESTART WITH DDL statements to resolve mismatches between source and target identity column values. No other actions are required.

Mismatch: XML DOCID sequence values

Db2 Cloning Tool Table Space Cloning automatically synchronizes target XML DOCID sequence values with their source counterparts. The source TCP/IP server job is required. If the target Db2 is a part of a data sharing group, Db2 Cloning Tool Table Space Cloning generates JCL jobs for updating the XML DOCID sequence values on other data sharing members. The generated JCL jobs should be run manually to update DOCID sequences. Also see the description of UPDATE-DOCID-JCL-DSN keyword in [“SET command and keyword definitions”](#) on page 568.

Mismatch: Index and index partition

All index mismatches can automatically be resolved by running a REBUILD utility. See the topic [Chapter 20, “Options for rebuilding indexes,”](#) on page 297 for more information.

Considerations for generating target object DDL using PROCESS-DDL

Db2 Cloning Tool Table Space Cloning can optionally generate and execute DDL to be used for creating non-existent or re-creating existing target objects. The **PROCESS-DDL** parameter of the **COPY** command generates DDL for storage groups, databases, table spaces, tables, indexes, views, aliases, synonyms, triggers, stored procedures, functions, data types, sequences, sequence aliases, global variables, and materialized query tables. LOB and XML spaces are supported. Cloning of archive tables is not supported. Db2 Cloning Tool Table Space Cloning always saves the generated source object DDL to a data set, or can run existing DDL from an input data set.

Note: If Db2 Cloning Tool Table Space Cloning modules on the target Db2 subsystem were bound with V12R1M504 application compatibility level or higher, Db2 Cloning Tool Table Space Cloning will automatically switch to V12R1M503 application compatibility level, so that deprecated object types can be created if necessary.

How LISTDEFs control DDL generation

As LISTDEF statements select objects to be processed, they also select the DDL to generate.

When missing target objects require DDL to be executed on the target, the DDL is built only for missing objects that must have a target data set for the copy from the source and objects they are dependent on.

When DDL for all source objects is to be created but not run, missing target objects are not relevant. In this case, LISTDEF is used differently to select objects that require DDL. DDL generation starts with all the table spaces that are directly or indirectly referenced in LISTDEF statements. This list of table spaces is referred to as the *object set*. All databases that contain table spaces in the object set have DDL generated. However, not all table spaces in each database have DDL generated. The table space must be in the object set selected by LISTDEF.

There must be at least one table space specified in the LISTDEF to generate DDL. A LISTDEF command with only indexes results in an error. All indexes must have a corresponding table space in the LISTDEF or be RI related (if specified). Indexes must also be specified in LISTDEF, either explicitly or implicitly (using **ALWAYS-COPY-INDEXSPACES(Y)**).

Note: Db2 Cloning Tool Table Space Cloning does not support the LISTDEF option ARCHIVE. Therefore, Db2 Cloning Tool Table Space Cloning does not generate DDL for archive tables, and cloning of archive tables is not supported.

DDL generation and object translation

Use object translation to change object names from the source to the target. Use **PROCESS-DDL** to generate and execute DDL to create missing or re-create existing target objects.

Whenever object translation is performed, Db2 Cloning Tool automatically generates **PROCESS-DDL** attribute change parameters. The generated syntax is included in the job log data set. For example: Assume the object translate card DATABASE ,X%,Y% results in the database name being translated from XMYDB to YMYDB. The **DDL-ATTRIBUTE-CHANGE** command will be generated as DDL-ATTRIBUTE-CHANGE(database ,XMYDB ,YMYDB ,% ,%) and results in every database string matching XMYDB changing to YMYDB in the generated DDL.

Recommendations



CAUTION: If you do not want DDL to be processed during a simulation, change **DDL-ENABLE** to N, or change **PROCESS-TYPE** to G and write the DDL to a file.

- When setting up a new clone operation, run the first time with **DATA-MOVER PGM(NONE)** to ensure the DDL is created and all the target objects are available for the copies. Then rerun the source job with **PGM** other than NONE, do the copies, and then run the target job. Once the source job is set up correctly, the DDL can be generated and the copies performed with one source job invocation.
- Specify all table spaces using a DATABASE or TABLESPACE object definition in LISTDEF and then specify **ALWAYS-COPY-INDEXSPACES(Y)**.

For additional information about the DDL processing, refer to the topics [“COPY” on page 513](#) and [“DDL-ATTRIBUTE-CHANGE parameter values” on page 555](#).

Re-creating existing target objects

Db2 Cloning Tool Table Space Cloning can optionally generate and run DDL to DROP and re-create existing target objects. All target objects can be re-created, or only the target objects that have mismatches can be re-created.

This feature is controlled by the RECREATE-OBJECTS keyword. It does not affect generating DDL for objects that do not exist on the target.

Re-creating all target objects

To re-create all target objects, specify RECREATE-OBJECTS(A). This will generate DROP and CREATE statements for all target objects.

Re-creating objects with mismatches

Db2 Cloning Tool Table Space Cloning can resolve object mismatches by dropping the target objects and re-creating them with the same attributes as the corresponding source objects. Re-creating objects can resolve most mismatches, with the following exceptions:

- CL_SOURCETYPEID. This type of mismatch cannot be automatically resolved.
- TB_VERSION, except when a source object does not have system pages and a target object has a non-zero version.
- TP_FORMAT. This type of mismatch can be resolved by running the REPAIR utility.
- TS_COMPRESS_TYPE. This type of mismatch can be resolved by running UNLOAD/LOAD utilities.
- TS_HASHSPACE. This type of mismatch can be resolved by running the REPAIR utility.
- Encryption mismatches. This type of mismatch can be resolved by running UNLOAD/LOAD utilities.

When RECREATE-OBJECTS(M) is specified and Db2 Cloning Tool encounters a mismatch for an object, its behavior depends on the setting for OBJECT-MISMATCH-RETURN-CODE for that mismatch, as follows:

- When OBJECT-MISMATCH-RETURN-CODE is set to 0, the mismatch is ignored.
- When OBJECT-MISMATCH-RETURN-CODE is set to 4, the object is marked for re-creation.
- When OBJECT-MISMATCH-RETURN-CODE is set to 8, the cloning ends with an error.

Restrictions for re-creating target objects

Objects are re-created in groups. A group of objects consists of a base table space with its tables, the indexes on those tables, XML and LOB spaces for the base table space and related indexes, and all RI-related objects with their indexes, XML spaces, and LOB spaces. To ensure that objects are properly re-created, Db2 Cloning Tool Table Space Cloning requires all objects from the same group to be included in the cloning. To enforce this restriction, specifying RECREATE-OBJECTS(A) or RECREATE-OBJECTS(M) requires the keywords ALWAYS-COPY-INDEXSPACES and INCLUDE-ALL-RI to be set to Y.

Before attempting to re-create an object, Db2 Cloning Tool Table Space Cloning checks the eligibility of the object and the objects from its group. If any re-create restriction is met for any object in the group, the entire group of objects cannot be re-created. A target object cannot be re-created under the following circumstances:

- The object is in UT* status.
- The source object resides in a simple table space.
- The object requires the use of user-managed data sets.
- The object is restricted for DROP, and the Db2 Cloning Tool Table Space Cloning COPY command keyword RECREATE-TABLE-RESTRICTED-ON-DROP is set to N.

Additional objects that exist on the target and do not exist on the source (for example, indexes or views) are not re-created.

When EXPLODE-OBJECTS is set to Y, the DDL for some types of objects (storage groups and data types) is always generated. It is recommended to specify the IGNORE-CREATE-OBJECT-EXISTS(Y) keyword to ignore SQLCODE=-601.

REMOTE-CONNECT-TYPE considerations

When connection to the target Db2 subsystem is via TCP/IP, and the RECREATE-OBJECTS parameter is set to A or M, then the target TCP/IP server job must include a CKZDDL DD. This DD must point to the sequential data set or PDS/PDSE member with RECFM=FB and LRECL=80. It must not point to the same data set or member as the PROCESS-DDL-DDN DD in the source job.

Recommendations for re-creating target objects

- When you want to use RECREATE-OBJECTS, run the Db2 Cloning Tool Table Space Cloning source job with PROCESS-TYPE(G), SIM(A), and, when applicable, WARN-IF-OBJECT-NOT-TRANSLATED(Y). Review all warnings in the source job output and the generated DDL before running the cloning with SIM(N).
- If target objects have associated history tables that are not part of the cloning, attempting to drop the target objects results in SQLCODE=-699. To prevent this issue, either include associated history tables in the cloning, or manually drop the history tables from the target subsystem before running the source job.

Sample DDL processing scenarios

This topic contains several scenarios for processing DDL with Db2 Cloning Tool Table Space Cloning.

Generating and running target DDL when not all target objects exist

In this scenario, some target objects do not exist and copies of all the source objects are to be made. Settings for Db2 Cloning Tool Table Space Cloning to generate and run DDL for non-existent target objects and then copy the source objects to the target are as follows:

- PROCESS-TYPE(Y)
- PROCESS-DDL-DDN(*ddname*)
- PGM(ADRSSU)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.

Changing the generated target DDL before running

In this scenario, you want to make changes to the Db2 Cloning Tool Table Space Cloning-generated DDL before running it. Generate the DDL in the first source job (using PGM(NONE) to prevent copies being made). Then run the DDL in the second source job.

Settings to generate DDL but not execute it, without copying the source objects to the target are as follows:

- PROCESS-TYPE(G)
- PROCESS-DDL-DDN(*ddname*)
- PGM(NONE)
- DEFVCAT(*vcatvalue*)
- COPY-IF-NO-DB2-TARGET-OBJECTS(Y)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.

Edit the DDL generated in *ddname*. Then rerun Db2 Cloning Tool Table Space Cloning to execute the previously written and edited DDL. No copies will be performed. The DDL can also be submitted by SPUFI or batch processing, outside of Db2 Cloning Tool Table Space Cloning. Once the target objects are complete, run Db2 Cloning Tool Table Space Cloning to make the copies.

- PROCESS-TYPE(X)
- PROCESS-DDL-DDN(*ddname*)
- PGM(NONE)

Use a source job with DLL-ENABLE(N) to copy objects from source to target.

Running generated target DDL when some target objects already exist

In this scenario, you have DDL for all of the target objects. However, some target objects already exist, but others do not. Settings for Db2 Cloning Tool Table Space Cloning to execute already generated DDL and copy the source objects to the target are as follows:

- PROCESS-TYPE(X)
- PROCESS-DDL-DDN(*ddname*)
- IGNORE-CREATE-OBJECT-EXISTS(Y)
- PGM(NONE)

Use a source job with DDL-ENABLE(N) to copy objects from source to target.

Generating DDL for the source objects

In this scenario, you want a copy of the source DDL. Settings for Db2 Cloning Tool Table Space Cloning to generate the source DDL without executing it:

- PROCESS-TYPE(A)
- PROCESS-DDL-DDN(*ddname*)
- PGM(NONE)
- Do not specify OBJECT-TRANSLATE or DDL-ATTRIBUTE-CHANGE commands, and the DDL will reflect the source objects.

Generating DDL for the target objects

In this scenario, you want a copy of all target DDL, regardless of whether the object exists on the target. Settings for Db2 Cloning Tool Table Space Cloning to generate target DDL without executing it are as follows:

- PROCESS-TYPE(A)
- PROCESS-DDL-DDN(*ddname*)
- PGM(NONE)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.
- Execute this source job using the old target subsystem as the LOCAL-SSID.

Generating and running target DDL for re-creating all target objects



Attention: If object translation rules are incorrectly specified, using RECREATE-OBJECTS(A) can drop and re-create the wrong objects. It is strongly recommended to first run the cloning job with PROCESS-TYPE(G), PGM(NONE), SIM(A), and, when applicable, WARN-IF-OBJECT-NOT-TRANSLATED(Y) to ensure that object translation is correct before running the main cloning job.

In this scenario, all target objects are dropped and re-created using source objects as models, so that all non-existent target objects are created. Settings for Db2 Cloning Tool Table Space Cloning to generate and run DDL and then copy the source objects to the target are as follows:

- PROCESS-TYPE(Y)
- RECREATE-OBJECTS(A)
- PROCESS-DDL-DDN(*ddname*)
- ALWAYS-COPY-INDEXSPACES(Y)
- INCLUDE-ALL-RI(Y)
- PGM(ADRDUSSU)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.

Generating and running target DDL for resolving object mismatches



Attention: If object translation rules are incorrectly specified, using RECREATE-OBJECTS(M) can drop and re-create the wrong objects. It is strongly recommended to first run the cloning job with PROCESS-TYPE(G), PGM(NONE), SIM(A), and, when applicable, WARN-IF-OBJECT-NOT-TRANSLATED(Y) to ensure that object translation is correct before running the main cloning job.

In this scenario, objects with mismatches are re-created and non-existent target objects are created. Settings for Db2 Cloning Tool Table Space Cloning to generate and run DDL and then copy the source objects to the target are as follows:

- PROCESS-TYPE(Y)
- RECREATE-OBJECTS(M)

- PROCESS-DDL-DDN(*ddname*)
- ALWAYS-COPY-INDEXSPACES(Y)
- INCLUDE-ALL-RI(Y)
- PGM(ADRDSSU)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.

Considerations for Db2 Version 9.1 and later

The following considerations apply when using Db2 Cloning Tool Table Space Cloning to clone table spaces and index spaces on a Db2 Version 9.1 or later subsystem.

Support for clone tables

Db2 Version 9.1 introduced the clone table, which is a copy of a table that resides in the same table space and that has the same attributes, structure, and data as the original base table. After you create a clone table, you can insert or load data into the clone table and exchange the clone table name with the base table name.

If a table is cloned, Db2 Cloning Tool Table Space Cloning always copies the clone table with its base table. Db2 Cloning Tool Table Space Cloning does not allow a base table or its clone to be copied to a table space that is not cloned, nor does it allow you to copy a non-cloned table to a base or cloned table space.

If the table space or index space does not exist on the target subsystem, Db2 Cloning Tool Table Space Cloning can copy both the base table and its clone table. Db2 Cloning Tool Table Space Cloning assigns the instance numbers in the underlying target VSAM data sets as F0001 for base tables, and F0002 for clone tables.

If desired, you can use the Db2 Cloning Tool Table Space Cloning LISTDEF command to selectively copy either only tables that have been cloned, or only those tables that have not been cloned.

Support for expanded index page sizes

Db2 Version 9.1 offers expanded index page sizes of 8 KB, 16 KB, and 32 KB. Db2 Cloning Tool Table Space Cloning can copy these expanded index page sizes.

Considerations for partition-by-growth table spaces

Db2 Version 9.1 introduced partitioning according to data growth. This enables segmented tables to be partitioned as they grow. Db2 Cloning Tool Table Space Cloning can copy partition-by-growth table spaces when the number of source partitions is equal to the number of target partitions. If the number of source partitions is greater than the number of target partitions, the target table space is automatically altered to add partitions. If the number of source partitions is less than the number of target partitions, specify the Db2 Cloning Tool Table Space Cloning UNLOAD-LOAD functionality to copy the table space.

Considerations for reordered row format in Db2 Version 9.1 NFM and later

Reordered row format (RRF) is new with Db2 Version 9.1 NFM. It results in variable-length columns being placed at the end of a row. RRF is not compatible with objects that have been migrated from Db2 V8. Those objects will be in basic row format (BRF) until a REORG or LOAD REPLACE is run on the object under Db2 Version 9.1 NFM.

Newly created table spaces under Db2 Version 9.1 NFM are put in RRF. If a source BRF table space is copied to a target RRF table space (or vice versa), in most cases the target table(s) will not be accessible. Db2 Cloning Tool Table Space Cloning will issue a warning if the row formats do not match. Thus, if running Db2 Cloning Tool Table Space Cloning on Db2 Version 9.1 NFM or later subsystem, and new objects are added to LISTDEF, run the source job using DATA-MOVER(NONE) and SIM(N). All object

incompatibilities will result in warning and/or error messages. REORG the table spaces in BRF and rerun the Db2 Cloning Tool Table Space Cloning source job to ensure there are no more object incompatibilities.

Considerations for data set renaming

This topic discusses considerations for renaming data sets.

Qualifiers cannot be added or removed

When cloning Db2 table spaces and index spaces, qualifiers cannot be added or removed because of Db2 naming standard requirements. If a new database exists on the target, the new database name will be used in the VSAM data set.

Target data set ICF catalog aliases

Users are responsible for creating ICF catalog aliases, if needed, for new target data set names. Db2 Cloning Tool Table Space Cloning gets the target high level qualifier from the VCAT in the Db2 catalog. There must be an ALIAS in the ICF catalog to match each VCAT in the Db2 catalog to enable the rename and catalog.

If the target table spaces and index spaces do not exist, then there must also be an ALIAS to match the DEFVCAT specified in the COPY command.

Considerations for using FUZZY-COPY

This topic discusses considerations for cloning data sets using FUZZY-COPY.

If the COPY command keyword FUZZY-COPY(Y) is specified, then Db2 Cloning Tool Table Space Cloning will not stop the source table spaces and index spaces. If the table spaces and index spaces are in read-write (RW) status, using FUZZY-COPY will not guarantee consistent data on the target without also using LOG-APPLY, as there might be uncommitted updates to the source data.

To clone using FUZZY-COPY(Y) without using LOG-APPLY, do the following:

1. Start all table spaces to be copied in read only (RO) status.
2. Run a QUIESCE with WRITE(YES) on all table spaces to be copied.
3. Run the source job.
4. Start all table spaces in read-write (RW) status.

FUZZY-COPY(Y) can be used with LOG-APPLY without these additional steps.

Considerations for using catalog prefetch to populate the object cache

Normally Db2 Cloning Tool Table Space Cloning reads the catalog using SELECTs with a WHERE clause. However, when indexes are not usable, one SELECT can take several seconds or longer. The CATALOG-PREFETCH command allows the rows in SYSIBM.SYSTABLESPACE, SYSIBM.SYSTABLEPART, SYSIBM.SYSTABLES, and SYSIBM.SYSINDEXES to be read in one pass for each database that has objects to be cloned. SYSIBM.SYSCOLUMNS and SYSIBM.SYSINDEXPART are read for each table and index that will be cloned.

If the source job is taking a long time to run, try REORGing the catalog. If REORGing the catalog is not an option, then using CATALOG-PREFETCH might be considered.

CATALOG PREFETCH should be used whenever the Db2 catalog on the source or target subsystem is unable to use indexes due to a poor cluster ratio.

Db2 Cloning Tool Table Space Cloning saves the prefetched objects based on the database list that is generated through LISTDEF processing.

Using the target prefetch database list can reduce the fetch time for the target catalog. This command allows all database objects to be saved in a single pass of the catalog tables being saved. When using a TCP/IP server with KEEP-DATABASES-ON-DISCONNECT(Y), the target database list command can specify a list of databases to be used by multiple source jobs. Once the cache is populated in the TCP/IP server, no further catalog accesses for the cached tables are required during target catalog lookup.

Run the initial source job to populate the TCP/IP server cache with PGM(NONE) and ENABLE-TARGET-PREFETCH(Y), and specify all the databases that will be used for cloning in one or more source jobs using TARGET-PREFETCH-DATABASE-LIST. Then run each source copy job (one at a time) using READ-FROM-SERVER-CACHE(Y) and ENABLE-TARGET-PREFETCH(N) to access the cache in the TCP/IP server. If the cache needs to be refreshed, re-run the source job with ENABLE-TARGET-PREFETCH(Y) and PGM(NONE).

When first setting up to copy a set of objects, run with different options and compare the elapsed times of the two jobs. Specify PGM(NONE) to prevent copies and space stops and starts when running the comparison jobs. See CKZ00302I and CKZ00303I messages in the source job log data set for statistics on catalog prefetch performance.

CATALOG-PREFETCH has the following requirement: Whenever the target objects do not exist or must be re-created, ENABLE-TARGET-PREFETCH(N) should be used. This includes objects that will be created or re-created with Db2 Cloning Tool Table Space Cloning DDL generation, and partition-by-growth partitions that do not exist on the target Db2 subsystem. When the target objects exist, specify ENABLE-TARGET-PREFETCH(Y) to populate the cache for target objects. Then, to read the objects from the cache without refreshing it, specify ENABLE-TARGET-PREFETCH(N) with READ-FROM-SERVER-CACHE(Y).

When using CATALOG-PREFETCH, only a single subtask is used to access the catalog. If you specify a subtask value greater than one, that value is used for additional (non-CATALOG-PREFETCH) processing in the source job, and is also used for the target job.

Note that not all catalog tables are cached. Among those that are accessed without using the cache are SYSDATABASE, SYSKEYS, SYSFORIGNKEYS, and SYSSEQUENCES. Access to these is limited and should not substantially impact source job performance.

Considerations for caching indexes

There are several considerations when planning to cache indexes using catalog prefetch.

If any indexes are present in a LISTDEF or ALWAYS-COPY-INDEXSPACES(Y) is specified, an attempt will be made to cache entries in SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART. The object specification for index space type specifications in the LISTDEF must be database or table space, just as it is for table space type specifications. LISTDEF statements for index space type specifications must follow all table space LISTDEF type specification statements.

Indexes that are not in the databases selected with table space caching, and index spaces that are cached using a specification other than database or table space type specifications, will not be cached. Therefore, indexes in index only jobs will not be cached, as there are no table space type specification statements.

The simplest way to ensure all indexes are cached is to only specify table space type specifications using database or table space object definitions in LISTDEF, and then use ALWAYS-COPY-INDEXSPACES(Y). If the COPY attribute is needed, then individual index statements must be specified with the COPY attribute added.

Note that when using ALWAYS-COPY-INDEXSPACES(Y), the internally created object spec is the same as it is for the corresponding table space type specification.

To include all index spaces in a database using COPY NO (the default), specify the LISTDEF and command as follows:

LISTDEF statements:

```
INCLUDE TABLESPACES DATABASE you1db1 ALL
INCLUDE TABLESPACES DATABASE you1db2 ALL
INCLUDE TABLESPACES DATABASE you1db3 ALL
```

COPY command:

```
ALWAYS-COPY-INDEXSPACES(Y)
```

To include all index spaces in a database using COPY YES (not the default), specify the LISTDEF as follows:

```
INCLUDE TABLESPACES DATABASE yourdb1 ALL
INCLUDE TABLESPACES DATABASE yourdb2 ALL
INCLUDE TABLESPACES DATABASE yourdb3 ALL
INCLUDE INDEXSPACES COPY YES DATABASE yourdb1 ALL
INCLUDE INDEXSPACES COPY YES DATABASE yourdb2 ALL
INCLUDE INDEXSPACES COPY YES DATABASE yourdb3 ALL
```

Using a local TCP/IP server to populate the target cache for multiple source copy jobs

This topic describes the procedure to use a local TCP/IP server to populate the target cache for multiple source copy jobs.

About this task

There are three copy jobs to be run.

Procedure

1. Specify SET KEEP-DATABASES-ON-DISCONNECT(Y) in the TCP/IP job parms.
2. Start the TCP/IP server on the LPAR used by the target subsystem.
3. In a pre-copy source job, specify:
 - a) PGM(NONE)
 - b) SET REMOTE-CONNECT-TYPE(T)
 - c) Add parms for catalog prefetch and use TARGET-PREFETCH -DATABASE-LIST to include the databases for all three jobs.
4. Run the job to populate the TCP server cache with all the target objects.
5. In all source copy jobs, specify:
 - a) SET REMOTE-CONNECT-TYPE(T)
 - b) Add parms for catalog prefetch.
 - c) Use the same TARGET-PREFETCH -DATABASE-LIST that was used in step 4.
 - d) PGM(ADRDSU) to do the copies.
6. Run source copy job 1.
7. Run source copy job 2.
8. Run source copy job 3.

Results

When steps 6, 7 and 8 are run, there will no DASD access for the three catalog tables that are cached to obtain target job information.

Return code choices

The CKZINI member in the product PARMLIB allows you to choose the seriousness of a return code. Scenarios discussed in this document assume that the conditional execution of subsequent steps adheres to the convention that return code 0 means successful, 4 means warning, and 8 means an error..

The following control parameters are in the product PARMLIB member CKZINI:

```
MAX_RC = 8           /* stop job when return code is greater than MAX_RC */
MAX_COPY_RC = 8     /* stop job when return code is greater than MAX_COPY_RC */
                    /* occurs when copying data */
```

These control parameters can be overridden by using the MAX-RC or the MAX-COPY-RC keywords of the SET command.

Plans and packages

Plans and packages must be bound on both the source and target Db2 systems before using Db2 Cloning Tool Table Space Cloning. All Db2 Cloning Tool Table Space Cloning packages are bound using package versions.

If you need to bind the plans, refer to [Chapter 4, “Customizing Db2 Cloning Tool,”](#) on page 77.

Dropped and altered tables

If any of the following is true, you might need to run a REORG on the source and/or target table spaces before copying them.

- One or more tables were dropped from the table space.
- If an ALTER TABLE ADD COLUMN was executed on the table space.
- If objects have been migrated from Db2 V7 and have had column(s) added using ALTER TABLE ADD COLUMN, note that the status AREO* may not be displayed. You must REORG the table spaces before using Db2 Cloning Tool Table Space Cloning to copy them. Failure to perform this REORG may result in the target data being inaccessible.

Copying versioned objects

Use the procedure in this topic when it is possible that one or more versioning ALTER commands have been done on any source or target table space, table, or index space in a Db2 Cloning Tool Table Space Cloning LISTDEF.

About this task

When an object is altered, Db2 can place versioning information in the catalog, Db2 directory, and the page set. If source and target version numbers do not match, it is important to ensure that after cloning completes, versioning information is consistent between the target page sets and the Db2 catalog. Db2 Cloning Tool Table Space Cloning can automatically detect and fix versioning mismatches by invoking the REPAIR utility for table spaces or the REBUILD INDEX utility for indexes.

Run the following procedure to ensure that the copied target objects are accessible:

Procedure

1. Run a REPAIR utility with keyword INSERTVERSIONPAGES on each source table space that does not have active versioning. This step ensures that all table spaces are self-describing. This step needs to be run only once.

Important: Db2 Epic 37738 (APAR PI86880 for Db2 12, APAR PI76179 for Db2 11) is required for the support of the INSERTVERSIONPAGES keyword in the REPAIR utility.

2. If the target Db2 subsystem is Version 11 and has all of the following APARs installed: PI78780, PI76461, PI76462, PI75145, PI81005, PI76179, and PI80006, set the CKZINI PARMLIB parameter DB2V11_EPIC37738_INSTALLED to Y.
3. Run the Db2 Cloning Tool Table Space Cloning source job with the following SET command parameters: TARGET-JOB-REPAIR-EXECUTE(Y) and REBUILD-INDEXES-EXECUTE(Y).
4. Run the Db2 Cloning Tool Table Space Cloning target job.

What to do next

If additional versioning incompatibilities are encountered that cannot be resolved automatically by Db2 Cloning Tool Table Space Cloning, review the messages that are returned in the job output (CKZ70803W, CKZ70804W, CKZ70805W, CKZ70807W, CKZ70808E, and CKZ70809W) for additional actions.

If no additional versioning incompatibilities are encountered that cannot be resolved automatically by Db2 Cloning Tool Table Space Cloning, no further action is required.

Chapter 13. Db2 Cloning Tool Table Space Cloning function overview

This topic offers a high level overview of the functions and jobs for using Db2 Cloning Tool Table Space Cloning.

Db2 Cloning Tool Table Space Cloning is comprised of the source job, the target job, and several optional jobs to facilitate cloning.

Source job overview

The Db2 Cloning Tool Table Space Cloning source job identifies all Db2 table spaces and index spaces to be copied to the target Db2 subsystem.

The source job identifies the spaces to be copied by:

- Selecting the table spaces and index spaces to be cloned via a LISTDEF facility, similar to the IBM LISTDEF
- Collecting information from both the source and target Db2 catalogs
- Deriving the names of the VSAM data sets for the source Db2 table spaces and index spaces
- Deriving the names of the VSAM data sets for the target Db2 table spaces and index spaces
- Issuing STOP commands for the target table spaces and index spaces (if requested by parms) and the source table spaces and index spaces
- Issuing the fast replication copy of the VSAM objects for FlashCopy or SnapShot using DFSMSdss or TimeFinder/Clone
- Generating the parameters and SQL that the target job uses to make the copied table spaces and indexes accessible at the target
- Writing the generated parameters and SQL to a sequential data set or PDS member which will be used as input for the target job that executes on the target Db2 z/OS system

The source job consists of 4 logical phases:

1. Init Phase, described in [“Source job - Init phase I”](#) on page 208.
2. Discovery Phase, described in [“Source job - Discovery phase II”](#) on page 208.
3. Eligibility Phase, described in [“Source job - Eligibility phase III”](#) on page 209.
4. Copy Phase, described in [“Source job - Copy phase IV”](#) on page 210.

Target job overview

The Db2 Cloning Tool Table Space Cloning target job makes the data accessible to the target Db2.

The target job makes the data accessible by:

- Translating the object IDs from the source table spaces and index spaces to those of the target table spaces and index spaces
- Resetting the LOGRBA on the target
- Optionally, starting the target table spaces and index spaces for RW (read/write) access

A target job repository can be defined and used to keep track of target jobs and the data sets that are processed by the target jobs. This repository allows the failed target job to be restarted, skipping any successfully processed target data sets or other successfully passed phases such as LOAD, RTS, DOCID update, or Db2 utilities. The USE-RUNTIME-REPOSITORY parameter of the SET command determines how the target job is processed. If set to N, the target job starts over. If set to Y, the target job skips

successfully passed phases and runs any failed phases and phases that were not run during the previous target job execution.

For details about using the runtime repository, see [“Runtime repository functionality overview”](#) on page 216.

TCP/IP server job overview (optional)

In the absence of a connection via CAF (Db2 Call Attach Facility), the Db2 Cloning Tool Table Space Cloning TCP/IP server job facilitates the communication of the Db2 Cloning Tool Table Space Cloning source job with a remote Db2 target subsystem.

Report job overview (optional)

The report job is a separate job that simply outputs the data in the target job runtime repository and ends.

Source job details

The source job identifies all Db2 table spaces and index spaces to be copied to the target Db2 subsystem using the Db2 Cloning Tool Table Space Cloning LISTDEF command, similar to IBM LISTDEF.

The default name of the source job in Db2 Cloning Tool Table Space Cloning is CKZxSRC, where x is a number 1-4 to match the setup scenarios in [Chapter 14, “Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone,”](#) on page 219 and [Chapter 15, “Set up procedures for copy by data set for all other methodologies,”](#) on page 237. The name for the target job is CKZTRG. Sample jobs can be found in the product JCL library.

Note: The source job should be scheduled when no utilities are running and there is no read/write activity against the table spaces and index spaces being copied.

The source job consists of four phases, documented in topics that follow.

Source job - Init phase I

The source job reads all Db2 Cloning Tool Table Space Cloning settings found in the product PARMLIB member CKZINI and in the CKZIN DD statement.

The CKZIN DD statement identifies the following source and target items:

- Source (local) and target Db2 subsystem names
- DDs passed to ADRDSSU or the EMC API for data set allocations
- IP address for TCP/IP (optional)
- A high level qualifier for the target VSAM data sets (if the Db2 table spaces and index spaces do not exist in the target Db2 catalog)
- Source/target creator ID pairs. A creator ID is the first qualifier of a Db2 table
- Copy options

Source job - Discovery phase II

The source job reads the Db2 Cloning Tool Table Space Cloning LISTDEF DD statement (CKZLSTDF) to determine the table spaces and index spaces to collect information from, in both the Db2 source and Db2 target catalogs. This phase determines what SQL queries are used to extract information from the source and target Db2 catalogs. No Db2 table spaces and index spaces are stopped during LISTDEF processing.

The Discovery Phase performs these tasks:

- Determines the source Db2 subsystem name.
- Determines all candidate source objects (databases, table spaces, tables, index spaces and indexes) to be used to determine data sets to be copied to the target Db2 subsystem based on the Db2 Cloning Tool

Table Space Cloning LISTDEF control statements. The user may optionally specify partitions of a partitioned table space or index space, RI or ALL (LOB specification).

- Eliminates any duplicate table spaces and index spaces specified in the LISTDEF command. Using the LISTDEF example that follows, all table spaces in database XXXXX are included in the copy except for those that begin with 'PRODAB. '. However, the third parameter is in error. The intent may have been to include one or more of the excluded table spaces. Instead it will override the exclude. Table spaces beginning with 'PRODA' will be duplicated. Db2 Cloning Tool Table Space Cloning will eliminate the duplicates caused by statement 1 and 3.

```
INCLUDE TABLESPACES DATABASE XXXXX
EXCLUDE TABLESPACES TABLESPACE XXXXX.PRODAB%
INCLUDE TABLESPACES TABLESPACE XXXXX.PRODA%
```

- Selects the list of candidate Db2 table spaces and index spaces from the source Db2 catalog and records:

Qualifiers (VCATs, creator IDs)
Object types (tables, indexes etc.)
Object names
Object attributes
Column attributes
Object IDs
and others

- Constructs the names of the VSAM data sets associated with the Db2 table spaces and index spaces.
- If LOB tables are present, Db2 Cloning Tool Table Space Cloning verifies that both the base tables and auxiliary table spaces are present for the copy and warns if one of them is missing.

Source job - Eligibility phase III

This phase confirms that a compatible target table space or index space exists on the target Db2 subsystem. Comparison of some of the attributes between the source and target table space or index space are made to determine if the data set can be copied from the source to the target.

For example, DSSIZE, page size (buffer pool), parted, CCSID, etc., are compared. Any attributes that may cause a conflict between the content or representation of the data in the source and target VSAM objects are compared.

Target analysis

- Uses TARGET_DB2(SSID(xxxx)) from the Db2 Cloning Tool Table Space Cloning COPY command to determine the target Db2 subsystem
- Uses CAF or TCP/IP to connect to the target Db2 subsystem
- Pairs the proper target creator ID with each source creator ID
- Verifies the existence of table spaces and index spaces on the target that match the candidate table spaces and index spaces on the source. If matching table spaces and index spaces are found, Db2 Cloning Tool Table Space Cloning compares:
 - Qualifiers
 - Object names
 - Object attributes
 - Column attributes
 - Object IDs
 - Buffer pools
 - CCSID
 - and others

- If no matching table space or index space is found and the COPY command keyword COPY-IF-NO-DB2-TARGET-OBJECTS(Y) is in effect, Db2 Cloning Tool Table Space Cloning:
 - Constructs a VSAM data set name based on the information for the source Db2 table space or index space
 - Uses the DEFVCAT qualifier as the HLQ for the VSAM data set name
 - Records its object IDs as zeroes
- If no matching table space or index space is found and the COPY command keyword COPY-IF-NO-DB2-TARGET-OBJECTS(N) is in effect, Db2 Cloning Tool Table Space Cloning sends a warning (CC 4) and records the missing table space or index space name(s) in CKZPRINT.
- If the source Db2 table space or index space is a partitioned table space or index space, Db2 Cloning Tool Table Space Cloning extracts Db2 catalog information about all of the target partitions in a single pass.
- If one or more target table spaces or index spaces do not exist in the catalog and you choose to copy the VSAM data set by specifying the COPY-IF-NO-DB2-TARGET-OBJECTS(Y) keyword, then Db2 Cloning Tool Table Space Cloning will not be able to extract the target object IDs for those data sets. At that time the object IDs will not exist. After the Db2 table spaces and index spaces are created by you, the source job may be re-run with the COPY command keyword DATA-MOVER(PGM(NONE)). The source job will update the SYNCDDB2 data set with the missing object IDs.

Source job - Copy phase IV

This phase determines if FlashCopy, SnapShot, or TimeFinder/Clone will be used to replicate the table spaces and index spaces, or if another copy mechanism will be used and prepares the parameters and SQL necessary to make the VSAM objects accessible at the target Db2.

Db2 Cloning Tool Table Space Cloning will invoke FlashCopy or SnapShot (if available) using DFSMSDss, program ADRDSSU, or uses the EMC API to invoke EMC TimeFinder/Clone, or you can use any other copy mechanism to replicate or copy the list of candidate data sets identified in the Eligibility Phase.

A source and target VSAM object pair is passed to the COPY Phase if:

- The attributes of the Db2 table spaces and index spaces associated with the source and target data set pairs are compatible between the source and the target subsystems
- If the object is a LOB, both the base and auxiliary table spaces must be included in the copy selection
- All table or index column order, column attributes and lengths are identical between the source and target table spaces and index spaces
- The target table spaces and index spaces exist or the keyword is COPY-IF-NO-DB2-TARGET-OBJECTS(Y)
- The spaces to be copied are not in certain restricted or advisory states (if requested in the SET command)

Copy steps

1. Copy source table spaces and index spaces to the target Db2.
 - Using FlashCopy, SnapShot, or TimeFinder/Clone: Unless the optional FUZZY-COPY is specified, Db2 Cloning Tool Table Space Cloning deallocates the source VSAM objects from the Db2 Database Manager address space prior to the Db2 Cloning Tool Table Space Cloning invocation of DFSMSDss, program ADRDSSU, to execute FlashCopy, SnapShot, or prior to invoking the EMC API for TimeFinder/Clone.
 - Using all other copy methodologies: The user is responsible for the deallocation of the source VSAM objects, and the copy methodology.
2. Db2 Cloning Tool Table Space Cloning restarts the source table spaces and index spaces after the copy is successful.

3. Db2 Cloning Tool Table Space Cloning creates control parameters in a sequential data set or a user-specified member of a PDS to:
 - Translate object IDs from the source table spaces and index spaces to the target table spaces and index spaces to make them accessible
 - Optionally, reset the LOGRBA for the target table spaces and index spaces
 - Optionally, start the target Db2 table spaces and index spaces after they are made accessible
 - Optionally, apply data masking to target tables, if requested
4. Db2 Cloning Tool Table Space Cloning may create SQL in a sequential data set or a user-specified member of a PDS to update any identity column information on the target.
5. Db2 Cloning Tool Table Space Cloning passes the RESET-LOGRBA parameter to the target job. This option resets the RBAs for the target VSAM objects. The LOGRBA will always be reset if there are OBID changes to be made. The level ID is always reset. Resetting the level IDs prevents Db2 from rejecting the data set as too old or unknown.

Target job details

The Db2 Cloning Tool Table Space Cloning target job must be submitted on the same z/OS system as the target Db2 subsystem by the user or job scheduler. This job resolves any outstanding issues with the target Db2 table spaces and index spaces to make them accessible.

Target job steps

Execute the commands and SQL passed from the Db2 Cloning Tool Table Space Cloning source job to:

- Use SQL to update any necessary ID column data
- Execute object ID translation for target Db2 table spaces and index spaces
- Optionally, reset the LOGRBA
- Optionally, apply Db2 logs to target tables, if requested
- Optionally, start the Db2 table spaces and index spaces
- Optionally, apply data masking to target tables, if requested

Once this job ends, the cloned table spaces and index spaces are accessible to Db2.

TCP/IP server job details (optional)

The optional Db2 Cloning Tool Table Space Cloning TCP/IP server job facilitates communication between the Db2 Cloning Tool Table Space Cloning source or target job and a target or source Db2 subsystem on a different LPAR. This is only necessary when a TCP/IP connection is used for the target subsystem.

The Db2 Cloning Tool Table Space Cloning source and target jobs can act as a TCP/IP client job. When a CAF connection cannot be established between the source LPAR and the target Db2 subsystem and between the target LPAR and the source Db2 subsystem, Db2 Cloning Tool Table Space Cloning source and target jobs automatically invoke TCP/IP to communicate with the server job on the remote LPAR. TCP/IP server jobs are submitted by the user and run on both source and target z/OS systems. TCP/IP server jobs must be started before submitting the source or target job that requires the use of the Db2 Cloning Tool Table Space Cloning TCP/IP client. TCP/IP must be active on same LPARs as the source and target Db2 subsystems.

TCP/IP server job steps

The TCP/IP server job has only one step and executes the same program as the source job.

The SET command keywords TCP-SERVER-JOB(Y) and SOURCE-TCP-SERVER-JOB(Y) shift the behavior of the job to that of communications facilitator.

The Db2 Cloning Tool Table Space Cloning source job will attempt to connect to the target Db2 in the following order:

1. CAF
2. TCP/IP

If the REMOTE-CONNECT-TYPE keyword is included on the SET command, Db2 Cloning Tool Table Space Cloning attempts to connect using the specified connection type (CAF or TCP/IP). If the specified connection type fails, Db2 Cloning Tool Table Space Cloning does not attempt to connect via the other connection type.

The TCP/IP target server job, CKZTCPS, must run on the same z/OS system as the target Db2. The control parameters for the CKZIN DD of the target server job are:

```
SET LOCAL-SSID(trss) TCP-SERVER-JOB(Y)
```

where:

trss is the target Db2 subsystem

keyword TCP-SERVER-JOB(Y) causes standard Db2 Cloning Tool Table Space Cloning JCL using PGM=CKZ00500 to become a server job.

The TCP/IP source server job, CKZTCPSS, must run on the same z/OS system as the source Db2. The control parameters for the CKZIN DD of the source server job are:

```
SET LOCAL-SSID(srss) SOURCE-TCP-SERVER-JOB(Y)
```

where:

srss is the source Db2 subsystem

keyword SOURCE-TCP-SERVER-JOB(Y) causes standard Db2 Cloning Tool Table Space Cloning JCL using PGM=CKZ00500 to become a server job.

Other options that are specific to TCP/IP server jobs include the following:

- TCPIP-SERVER-PORT and SOURCE-TCPIP-SERVER-PORT specify the connection port.
- TCPIP-SERVER-STC-NAME and SOURCE-TCPIP-SERVER-STC-NAME specify the TCP/IP address space name.
- CONNECT-DB2-ON-CLIENT-CONNECT and SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT specify whether to disconnect from Db2 on TCP/IP disconnect.
- IP-VERSION6 and SOURCE-IP-VERSION6 specify the IP version; specify N to indicate IPv4.

For more information about these commands, see the topic [Chapter 27, “Db2 Cloning Tool Table Space Cloning commands,”](#) on page 513.

The TCP/IP server jobs use port 5099 as the default. The same port must be used by the client job. The parameters used by both the server job and the client job can be set using tokens in the TCPIP_OPTIONS section of *hlq1*.SCKZPARAM(CKZINI), as follows:

```
:TCPIP_OPTIONS
TCPIP_STC_NAME      = TCPIP          /* LOCAL TCPIP STC NAME
TCPIP_SERVER_PORT  = 5099           /* TCPIP SERVER PORT #
```

where:

TCPIP is the name of the local TCP/IP address space on the source side

5099 is the port on which the TCP/IP server is listening on the target side

hlq1 is the data set high level qualifier used for the Db2 Cloning Tool Table Space Cloning PARMLIB member. However, it is not required that the client and server jobs use the same PARMLIB member. If, for example, the name of the TCP/IP address spaces differ on the source and target z/OS systems, different PARMLIB members must be used. You must make certain that the port names and numbers in both the source and target PARMLIB members still match. Or, you can set the port names and numbers using the TCPIP-SERVER-PORT and TCPIP-STC-NAME parameters of the SET commands in the source or TCPIP jobs.

The IP address or DNS names of the source and target z/OS systems are required. These can be specified in SET or COPY commands of the source job. Support is available for both IPv4 and IPv6. To use IPv6, specify IP-VERSION6(Y) or SOURCE-IP-VERSION6(Y) on the TCP/IP server SET command. Then, if IP addresses are used to identify the source or target z/OS system, specify an IPv6 address (using : for the field separator) in the client SET or COPY command. If the source or target systems are identified by DNS name, specify the DNS name in the client SET or COPY command. The client job will resolve the DNS name to the correct IPv4 or IPv6 address.

The TCP/IP server jobs require an initiator and must be accessed by client jobs in series. That is, only one client can be connected to a specific server at any one time. To run another server on the same z/OS system, submit another job with a different port number and/or TCP/IP started task name.

If you are running multiple source or target jobs concurrently, you should have a different TCP/IP server job running for each source or target job. Each TCP/IP server and source or target job must use a unique TCP/IP port number. The same server job can be used if the source or target jobs are run in series.

A separate TCP/IP server is required for each Db2 subsystem that is accessed concurrently on a particular LPAR. Each TCP/IP server must have its own port for the client to connect to.

The TCP/IP server job can be run as a started task. When run as a started task, SYSCTL authority must be granted to the started task's user ID on the target Db2 subsystem.

Stopping the TCP/IP server job

Once submitted, the TCP/IP server job will run until canceled or the time limit is exceeded. It may be canceled via the standard JES2 cancel command:

```
C XXXXXXX
```

The TCP/IP server job also supports the MVS STOP command. When the server job receives an MVS STOP command, it continues to run until the current client disconnects and shuts down.

Report job details (optional)

The optional Db2 Cloning Tool Table Space Cloning report job produces a report of the contents of the target job runtime repository.

If you plan to run a report job, ensure that the repository data sets are allocated before the target job is run. Data set allocations are contained in the SCKZJCL library member CKZRREP.

The report job can provide a data set report and a job report.

Data set report

The data set report contains information such as the source data set name (in the long form) and the target data set names, the processing return code, and the number of changed pages. To generate a data set report, the CKZRRDSN repository DD must be specified with the corresponding repository data set. In addition, you must provide one or both of the following DDs:

- Report in short form (RECFM=FBA, LRECL=133):

```
//CKZDREPS DD DISP=SHR,DSN=&dsnshort(&mbr)
```

- Report in long form (RECFM=FBA, LRECL=201):

```
//CKZDREPL DD DISP=SHR,DSN=&dsnlong(&mbr)
```

If you want to generate both the long and the short report, specify both DDs.

Jobs report

The jobs report provides information about the target jobs that were run for the same source job, and contains information such as the execution date and time, the job name, and the source and target SSIDs. To generate a jobs report, the CKZRRJOB repository DD must be specified with corresponding repository data set. In addition, you must provide at least one of the following DDs:

- `//CKZJREPL DD DISP=SHR,DSN=&joblong(&mbtr)`
- `//CKZJREPS DD DISP=SHR,DSN=&jobshort(&mbtr)`

Both forms of the jobs report are the same. If both DDs are specified, the report will be printed only in CKZJREPL. If only one of the DDs is specified, the report will be printed in that DD. Both forms are RECFM=FBA, LRECL=133.

A sample report job is contained in the SCKZJCL library member CKZREPJB. This job prints both the data set and the jobs report.

Report layout

The following tables describe the columns in the job report and the data set report.

| <i>Table 44. Columns on both short and long form of the job report</i> | |
|--|--|
| Column | Description |
| type | This column contains RRJB. |
| src job date | The date the source job was started. |
| src job time | The approximate time the source job started. |
| src job name | The source job name. |
| src job number | The source job JES number. |
| nr | The Iteration of the target job: 01 if only run once. There is a row in the report each time the target job is run or rerun. |
| trg system | The target z/OS system id. |
| trg job date | The date of the target job was started. |
| trg job time | The approximate time the target job was started. |
| trg job name | The target job name. |
| trg job number | The source job JES number. |
| src SSID | The Db2 source subsystem. |
| trg SSID | The Db2 target subsystem. |
| trg job RC | The target job return code. |
| trg job RS | The target job reason code. |

| <i>Table 45. Columns on the short form of the data set report</i> | |
|---|--------------------------------------|
| Column | Description |
| type | This column contains RRDS. |
| trg job date | The date the target job was started. |

Table 45. Columns on the short form of the data set report (continued)

| Column | Description |
|------------------------------|---|
| trg job time | The approximate time the target data set processing started |
| trg data set name | The name of the target data set. |
| data set RC | The two-digit data set return code. |
| elapsed time in seconds | The elapsed time to process the data set in seconds. |
| space type | The space type, either tablespace or indexspace. |
| clone | This column is blank if the data set is not a clone data set. If the data set is a clone, the column contains B if a clone base and C if a clone. |
| # log pages changed | The number of pages that had one or more log records applied. |
| # data masking pages changed | The number of pages that had one or more data masking changes applied. |
| # VSAM reads | The total number of VSAM reads, including zero pages. |
| # VSAM writes | The total number of changed pages. |

Table 46. Columns on the long form of the data set report

| Column | Description |
|------------------------------|---|
| type | This column contains RRDS. |
| trg job date | The date the target job was started. |
| trg job time | The approximate time the target data set processing started. |
| src data set name | The name of the source data set. |
| trg data set name | The name of the target data set. |
| data set RC | The two-digit data set return code. |
| pages | The total number of pages in this data set. |
| elapsed time in seconds | The elapsed time to process the data set, in seconds. |
| space type | The space type, either tablespace or indexspace. |
| clone | This column is blank if the data set is not a clone data set. If the data set is a clone, the column contains B if a clone base and C if a clone. |
| # log pages changed | The number of pages that had one or more log records applied. |
| # data masking pages changed | The number of pages that had one or more data masking changes applied. |

Table 46. Columns on the long form of the data set report (continued)

| Column | Description |
|---------------|---|
| # VSAM reads | The total number of VSAM reads, including zero pages. |
| # VSAM writes | The total number of changed pages. |
| IO err | This column contains Y if an I/O error occurred. |
| VSAM err | This column contains Y if a VSAM error occurred. |

CKZFIX job details (optional)

The optional Db2 Cloning Tool Table Space Cloning fix job, CKZFIX, facilitates dealing with missing target Db2 table spaces and index spaces.

The CKZFIX job uses parameters automatically generated by the source job to make the data for the missing table spaces and index spaces available to Db2. Sample JCL for this job is in the product JCL library. You must create the missing Db2 table spaces and index spaces either manually or via a tool and then run the fix job. CKZFIX must execute on the target z/OS system.

CKZFIX job steps

1. Stop: Stops the newly created (formerly missing) Db2 table spaces and index spaces. This deallocates them from Db2. This requires a resubmittal of the source job after the table spaces and index spaces are created.
2. Delete/ rename: Deletes the VSAM data set created by Db2 and renames the replicated VSAM data set to take the place of the deleted data set.

CKZSTPT, CKZSTPS and CKZSTRS jobs

The optional CKZSTPT, CKZSTPS and CKZSTRS jobs facilitate in replicating Db2 data sets outside of Db2 Cloning Tool Table Space Cloning. They stop the target Db2 table spaces and index spaces, stop the source Db2 table spaces and index spaces and restart the source Db2 table spaces and index spaces.

These jobs are user submitted. At the user's request, Db2 Cloning Tool Table Space Cloning will generate parameter for these jobs. The JCL for these jobs is in the product JCL library.

- CKZSTPT has only one step. This job submits Db2 commands to stop the target Db2 table spaces and index spaces and to deallocate the target VSAM data sets. It must execute on the target z/OS system.
- CKZSTPS has only one step. This job submits Db2 commands to stop the source Db2 table spaces and index spaces and to deallocate the source VSAM data sets. It must execute on the source z/OS system.
- CKZSTRS has only one step. This job submits Db2 commands to start the source Db2 table spaces and index spaces after they have been copied to the target.

Runtime repository functionality overview

The runtime repository functionality makes target job restartability possible. The runtime repository data sets store information about the executed phases of the target jobs and the processed data sets.

A pair of runtime repository data sets (*prefix*.RRJOB and *prefix*.RRDSN) can be used for target jobs that are related to only one and the same source job. The source job does not use runtime repository data sets; they are used only in the target job.

The following phases of the target job are restartable:

- LOAD utility processing. If the SYSPUNCH data set was processed successfully, it is not processed again during the target job restart.

- If USE-LAST-CONSISTENT-FLASHCOPY was in use in the source job, and an IDCAMS call was required to rename data sets, IDCAMS is not called again during the target job rerun, as long as during the previous run all of the IDCAMS ALTERs completed successfully.
- Data set processing. If an object for the data sets was required to be stopped and was successfully stopped, then a STOP command is not again issued against that object. If data set page processing finished successfully, it is not processed again for that data set during the target job restart. If an object was started successfully, then a START command will not again be issued against that object.
- RTS data set processing. If an RTS data set was processed successfully, it is not processed again during the target job restart.
- XML DOCID processing. If DOCIDs were updated on the current member, they are not updated again during the target job restart. If JCL for updating DOCIDs for other members of a data sharing group was written successfully, the JCL is not written again during the target job restart.
- REBUILD and REPAIR utilities. If one utility execution call finished successfully, the utility will not be executed again during the target job restart.

If target log apply functionality is used with END-POINT TO_CURRENT, during the first run of the target job TO_CURRENT is changed to TO_LOGPOINT. The TO_LOGPOINT VALUE is retrieved from the source subsystem BSDS data set after execution of the SET LOG LOGLOAD(0) command. This log point is used for all target job reruns to provide consistency of target objects. Minilog data sets should be deleted before each rerun of the target job.

SYNCDB2 should not be changed between target job executions. If changes are made to SYNCDB2, the runtime repository issues an error message during comparison of the RRDSN/RRJOB and SYNCDB2 data.

If RTS or LOAD functionality will be used with the runtime repository, it is recommended to set the parameters DELETE-RTS-DATASETS (for RTS) and DELETE-DATASETS (for LOAD) to N to avoid errors during the target job rerun.

The runtime repository functionality cannot be used during a simulation run. A warning message is issued if SIM(A) and USE-RUNTIME-REPOSITORY(Y) are both specified, and the runtime repository is disabled.

Chapter 14. Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone

This topic contains Db2 Cloning Tool Table Space Cloning setup procedures for you to use if data set copies are to be created with FlashCopy or SnapShot via DFSMSdss or TimeFinder/Clone.

Setup procedures for scenarios using methodologies other than FlashCopy, SnapShot, or TimeFinder/Clone are documented in [Chapter 15, “Set up procedures for copy by data set for all other methodologies,”](#) on page 237.

Setup for scenario 1: FlashCopy, SnapShot, or TimeFinder/Clone - All target table spaces and index spaces already exist in Db2 catalog

Use the procedure in the following table when FlashCopy, SnapShot, or TimeFinder/Clone is used to copy source table spaces and index spaces to the target Db2 subsystem when the target table spaces and index spaces already exist in the Db2 catalog.

| |
|---|
| <i>Table 47. Db2 Cloning Tool Table Space Cloning - Setting up for using FlashCopy, SnapShot, or TimeFinder/Clone when all target objects already exist</i> |
| FlashCopy, SnapShot, or TimeFinder/Clone Setup Steps |
| “Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog” on page 219. |
| “Step 2: Set up and execute source job (CKZ1SRC)” on page 220. |
| “Step 3: Set up and execute target job (CKZTRG)” on page 223. |

Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog

The first step when copying by data set using FlashCopy, SnapShot, or TimeFinder/Clone is to create the target objects in the target Db2 catalog.

Db2 associates Db2 table spaces and index spaces with specific VSAM data set names. For example:

```
VCAT.DSNDBD.DBTEST1.TESTTBSP.I0001.A001
```

where:

- Node 1: VCAT - the data set high level qualifier
- Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion
- Node 3: DBTEST1 - the database name in the Db2 catalog
- Node 4: TESTTBSP - the table space or index space name in the Db2 catalog
- Node 5: Either I0001 or J0001 (a switch indicator)
- Node 6: A001 to E094, a partition number or Db2 extent number

The correctly named Db2 table spaces and index spaces must exist in the target Db2 catalog for Db2 to recognize and use the VSAM data sets cloned to the target by Db2 Cloning Tool Table Space Cloning.

When Db2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets cloned from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. Db2 Cloning Tool Table Space Cloning

will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target Db2.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: Db2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up and execute source job (CKZ1SRC)

The source job is comprised of several commands and phases to control the selection of the Db2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

Before you begin

Note: It is recommended that you first run the source job with CHECK-DATASET-COMPATIBILITY(Y) in simulation mode (PGM(NONE), or SIM(A) if PGM(ADRDSSU) is specified) to determine whether the VSAM attributes of the source data sets and the target data sets are compatible. After you resolve any data set incompatibilities, then set CHECK-DATASET-COMPATIBILITY to N and run the source job with PGM and SIM set as desired. To avoid unnecessary CPU and I/O time, do not use CHECK-DATASET-COMPATIBILITY(Y) in a non-simulation run of the source job.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ1SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to Db2 Cloning Tool Table Space Cloning using the Db2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where Db2 Cloning Tool Table Space Cloning will build the input parameters for the target job.

See line #7 in the source job example and the associated note 7.
3. Identify the source Db2 subsystem in the SET command, keyword LOCAL-SSID(n), where n is the name of your source Db2 subsystem.

See line #8 in the source job example and the associated note 8.

4. If input and/or output DDs must be passed to ADRDSSU, use the HLQDDDF command to specify those DDs.

Identify the HLQ (high level qualifier) and a DD name or names using keyword HLQNAME(n) and DDNAME(dd1...,ddn), where n is your high level qualifier and dd1...ddn is one or more DDs. See line #9 in the source job example and the associated note 9.

5. Identify the target Db2 subsystem information and copy method in the COPY command.

- Identify the target Db2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(n) where n is the name of your target Db2 subsystem. See line #10 in the source job example and the associated note 11.
- If using data set level FlashCopy or SnapShot to copy table spaces and index spaces, and your environment is set up to support data set level replication, use COPY command keyword, DATA-MOVER(PGM(ADRDSSU)) which invokes DFSMSdss program ADRDSSU to execute either FlashCopy or SnapShot to do the data set level copy operation. If FlashCopy is not available, ADRDSSU will substitute a normal copy operation, unless REQUIRED is specified. See line #10 in the source job example and the associated note 10. If using EMC TimeFinder/Clone, specify COPY command keyword DATA-MOVER(PGM(EMCAPI)).

Results



CAUTION: If COPY command keyword FUZZY-COPY(Y) is specified, then Db2 Cloning Tool Table Space Cloning will not stop the source table spaces and index spaces. However, we *do not recommend* this, because if the table spaces and index spaces are in RW status, there are data integrity issues.

Source job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=srcsys
/**
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                            *
/*****
/**
/** Scenario 1 - all spaces to be copied exist on the target and
/**      CKZ controls the copy process
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** srcsys - name of source system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSMDUMP high level qualifier
/** dsnvrl - DB2 high level qualifier
/** parmbr - CKZ PARMLIB member
/** dumpsc - SYSMDUMP STORCLAS if needed
/** dumpmc - SYSMDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmb - LISTDEF member name
/** trss - target DB2 subsystem
/** dddd - DD DISP value if needed (must enable)
/** uuuu - DD UNIT value if needed (must enable)
/** vvvvvv - DD VOLUME value if needed (must enable)
/** srss - source DB2 subsystem
/** defsqlid - default SQLID applied to LISTDEF objects if needed
/** ipaddr - target IP if using TCPIP (delete if not needed)
/** ocs - object creator on source DB2 (delete if not needed)
/** oct - object creator on target DB2 (delete if not needed)
/**
/**
```

```

    /** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
    /**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
    DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
    DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
    /**
    /**
    /** CKZINI - Reads the CKZ parmlib to get default settings
    /**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmmbr)
    /**
    /**
    /** CKZPRINT - primary message output, required
    /**
4 //CKZPRINT DD SYSOUT=*
    /**
    /**
    /** CKZLOG - detailed message output, optional
    /**
5 //CKZLOG DD SYSOUT=*
    /**
    /**
    /** dump DDs
    /**
    /***SYSUDUMP DD SYSOUT=* Note: SYSDUMP preferred
    /***SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
    /** STORCLAS=dumpsc,MGMTCLAS=dumpmc,
    /** SPACE=(CYL,(50,30),RLSE),
    /** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
    /***ABNLIGNR DD DUMMY do not remove if using ABENDAID
    /**
    /**
    /** CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
    /** to select the source spaces to be copied to the target.
    /**
6 //CKZLSTDF DD DISP=SHR,DSN=hlq3.LISTDEF(lstdmbr)
    /**
    /**
    /** CKZSYNC - output commands that will be used by the target
    /** job to make the VSAM object(s) accessible on the target
    /** DB2 subsystem.
    /**
7 //CKZSYNC DD DISP=OLD,DSN=hlq3.SYNCDDB2(lstdmbr)
    /**
    /**
    /** DSS output DD if required, see HLQDDDF below
    /**
    /***MYOUTDD DD DISP=dddd,UNIT=uuuu,VOL=SER=(vvvvvv)
    /**
    /*******
    /** sample control statements
    /*******
    /**
    /**CKZIN DD *
    /**
    /**
    /** Source DB2 subsystem name and default SQLID if needed
    /**
    /**
8 SET LOCAL-SSID(ssid) DEFAULT-SQLID(defsqlid)
    /**
    /**
    /** CKZ sends this DD name to ADRDSSU as output (enable if needed)
    /**
    /**
9 /** HLQDDDF HLQNAME(trghlq) DDNAME(MYOUTDD)
    /**
    /**
    /**
    /** The COPY command invokes DFSMSdss program, ADRDSSU, to
    /** execute FlashCopy or SnapShot and prepares the object ID
    /** translation parameters for the target job. Or, if COPY
    /** keyword DATA-MOVER(PGM(NONE)) is present, CKZ outputs
    /** commands to assist the manual copy process.
    /**
    /**
10 COPY TARGET-DB2(SSID(trss) -
    IP(ipaddr) -
    SYNCDB2-DDN(CKZSYNC) -
    SIM(N) -
    DATA-MOVER(PGM(ADRDSSU) -
    FASTREP(PREF)) -
    COPY-IF-NO-DB2-TARGET-OBJECTS(N) -
    OBJECT-TRANSLATE(CREATOR,ocs,oct)-
    AUTO-STOP-TARGET-SPACE(Y) -

```

```
AUTO-START-TARGET-SPACE(Y)      -  
FUZZY-COPY(N)                   -  
RESET-LOGRBA(Y)
```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated Db2 table spaces and index spaces, Db2 start and stop space command status and DFSMSdss program ADRDSSU commands and status. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG - Displays LISTDEF processing and Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. CKZLSTDF - Db2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *
7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target Db2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. The SET LOCAL command specifies the local Db2 subsystem for the source job and the TCP/IP server job.
9. The optional HLQDDDF command may be used to pass input and/or output DDs to ADRDSSU. This may be useful to pass volsers to ADRDSSU for non-SMS output volume allocations.
10. The COPY command invokes the DFSMSdss program ADRDSSU to execute FlashCopy or SnapShot, or invokes the EMC API to execute TimeFinder/Clone, and prepares the object ID translation parameters for the target job. You may also specify the SIMULATE keyword with the COPY command.

Step 3: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and a SYNCDB2 command for each data set copied to make the VSAM objects accessible on the target subsystem. It will optionally start the target table spaces and index spaces. The target job assumes the table spaces and index spaces are already stopped.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. The sample target job is valid for all four scenarios described. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Specify the PARMLIB data set. Note that the Db2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCPIP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target, and TCP/IP server jobs.
See line #3 in the target job example and the associated note 3.
2. Specify the input data set where the Db2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, on line #6.
See line #6 in the target job example and the associated note 6.

Target job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The target job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=trgsys
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                            *
/*****
/**
/** Scenarios 1-4 - the target job is the same for all 4 scenarios.
/** The only change is the data set specified in CKZSQL and CKZIN.
/** These are output from the source job.
/**
/** The target job must be run on the target DB2 system.
/**
/** variables to be filled in ...
/**
/**      jobcard - job card
/**      trgsys  - name of target system
/**      hlq1   - CKZ LOADLIB and PARMLIB high level qualifier
/**      hlq2   - SYSDUMP high level qualifier
/**      dsnvrl - DB2 high level qualifier
/**      parmmb - CKZ PARMLIB member
/**      dumpsc - SYSDUMP STORCLAS if needed
/**      dumpmc - SYSDUMP MGMTCLAS if needed
/**      hlq3   - CKZ users high level qualifier
/**      lstdmb - LISTDEF member name
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1   EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
//          DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
//          DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI  DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmmb)
/**
/**
/** CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD SYSOUT=*
/**
/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG  DD SYSOUT=*
/**
/**
/** dump DDs
/**
/** *SYSUDUMP DD SYSOUT=*          Note: SYSDUMP preferred
/** *SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
/**          STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/**          SPACE=(CYL,(50,30),RLSE),
/**          DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/** *ABNLIGNR DD DUMMY            do not remove if using ABENDAID
/**
/**
/** CKZIN - input commands that will be used by the target
/** job to make the VSAM object(s) accessible on the target
/** DB2 subsystem. Created by the source job.
/**
6 //CKZIN   DD DISP=SHR,DSN=hlq3.SYNCDB2(lstdmb)
/**
/*****
/** sample control statements as would appear in SYNCDB2 member
/** Note: all lines have /* added in column 1 for JCL compatibility.
/*****
```

```

/**
/**CKZIN DD *
/** 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
/** SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
/**
/**SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCAN0(N)
/**
/**
/** TABLE SPACES - COPIED SUCCESSFULLY */
/**
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'012B,X'0113, -
/** PSID,X'0002,X'0047, -
/** TSOB,X'0001,X'0046, -
/** TBOB,X'0003,X'0048, -
/** TBOB,X'0008,X'004D, -
/** TBOB,X'000D,X'0052)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0024,X'0013, -
/** TSOB,X'0023,X'0012, -
/** TBOB,X'0025,X'0014, -
/** TBOB,X'002A,X'0019, -
/** TBOB,X'002F,X'001E)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0046,X'0035, -
/** TSOB,X'0045,X'0034, -
/** TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0114, -
/** PSID,X'0046,X'0035, -
/** TSOB,X'0045,X'0034, -
/** TBOB,X'0047,X'0036)

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT - Displays CKZINI tokens, CKZIN control parameters, Db2 SQL execution status, and SYNCDB2 status and START Db2 command status for each data set processed. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.

5. CKZLOG - Displays the Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each Db2 page access. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Setup for scenario 2: FlashCopy, SnapShot, or TimeFinder/Clone - Some or all target table spaces and index spaces do not exist in Db2 catalog

Ideally, all target Db2 table spaces and index spaces will exist before the Db2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy and there is not enough time to fit creating the target table spaces and index spaces within the window, then use this procedure.

You may also use the procedure (shown in the following table) for the table spaces and index spaces that are inadvertently missed. The Db2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. Only the data sets missing on the target require CKZFIX to be run.

| |
|--|
| <i>Table 48. Db2 Cloning Tool Table Space Cloning - Setting up for using FlashCopy, SnapShot, or TimeFinder/Clone when some or all target objects do not exist</i> |
| FlashCopy, SnapShot, or TimeFinder/Clone Setup Steps: when some or all target objects do not exist |
| “Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog (if required)” on page 226. |
| “Step 2: Set up and execute source job (CKZ2SRC)” on page 227. After you submit the source job and Db2 Cloning Tool Table Space Cloning detects missing target table spaces and index spaces, Db2 Cloning Tool Table Space Cloning will: <ul style="list-style-type: none"> • Modify the target data set name(s) that it creates (the first character of the fifth node will be changed from “I” or “J” to “F”. Db2 will not recognize the new data set name.) • Generate IDCAMS parameters to delete the VSAM object(s) which will, later, be created by the target Db2. • Generate IDCAMS parameters to rename the “F” data sets to a name recognizable to Db2. |
| “Step 3 (Optional): Create missing target table spaces and index spaces” on page 231. |
| “Step 4: Set up and execute optional FIX job (CKZFIX)” on page 231. |
| “Step 5: Set up for a re-run of the source job” on page 232. |
| “Step 6: Set up and execute target job (CKZTRG)” on page 233. |

Step 1: Create target Db2 table spaces and index spaces in the target Db2 catalog (if required)

The first step when copying by data set using FlashCopy, SnapShot, or TimeFinder/Clone when some or all target table spaces and index spaces do not exist is to create the target objects.

Db2 associates Db2 table spaces and index spaces with specific VSAM data set names. For example:

```
VCAT.DSNDBD.DBTEST1.TESTTBL.I0001.A001
```

where:

- Node 1: VCAT - the data set high level qualifier
- Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion
- Node 3: DBTEST1 - the database name in the Db2 catalog
- Node 4: TESTTBL - the table space or index space name in the Db2 catalog
- Node 5: Either I0001 or J0001 - a switch indicator
- Node 6: A001 to E094A - a partition number or Db2 extent number

The correctly named Db2 table spaces and index spaces must exist in the target Db2 catalog for Db2 to recognize and use the VSAM data sets cloned to the target by Db2 Cloning Tool Table Space Cloning.

When Db2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets copied from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. Db2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target Db2. Therefore you must create those target table spaces and index spaces if they do not exist.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: Db2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up and execute source job (CKZ2SRC)

The source job is comprised of several commands and phases to control the selection of the Db2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

Before you begin

Note: It is recommended that you first run the source job with CHECK-DATASET-COMPATIBILITY(Y) in simulation mode (PGM(NONE), or SIM(A) if PGM(ADRDSSU) is specified) to determine whether the VSAM attributes of the source data sets and the target data sets are compatible. After you resolve any data set incompatibilities, then set CHECK-DATASET-COMPATIBILITY to N and run the source job with PGM and SIM set as desired. To avoid unnecessary CPU and I/O time, do not use CHECK-DATASET-COMPATIBILITY(Y) in a non-simulation run of the source job.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ2SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to Db2 Cloning Tool Table Space Cloning using the Db2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where Db2 Cloning Tool Table Space Cloning will build the input parameters for the target job.

See line #7 in the source job example and the associated note 7.
3. Identify the source Db2 subsystem in the SET command, keyword LOCAL-SSID(n), where n is the name of your source Db2 subsystem.

See line #8 in the source job example and the associated note 8.
4. If input and/or output DDs must be passed to ADRDSSU, use the HLQDDDF command to specify those DDs.

Identify the HLQ (high level qualifier) and a DD name or names using keyword HLQNAME(n) and DDNAME(dd1...,ddn), where n is your high level qualifier and dd1...ddn is one or more DDs. See line #9 in the source job example and the associated note 9.
5. Identify the target Db2 subsystem information and copy method in the COPY command.
 - Identify the target Db2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(n) where n is the name of your target Db2 subsystem. See line #10 in the source job example and the associated note 10.
 - If using data set level FlashCopy or SnapShot to copy table spaces and index spaces, and your environment is set up to support data set level replication, use COPY command keyword, DATA-MOVER(PGM(ADRDSSU)) which invokes DFSMSdss program ADRDSSU to execute either FlashCopy or SnapShot to do the data set level copy operation. If FlashCopy is not available, ADRDSSU will substitute a normal copy operation, unless REQUIRED is specified. See line #10 in the source job example and the associated note 10. If using EMC TimeFinder/Clone, specify COPY command keyword DATA-MOVER(PGM(EMCAPI)).
 - Specify COPY parm COPY-IF-NO-DB2-TARGET-OBJECTS(Y) to prevent Db2 Cloning Tool Table Space Cloning from issuing a warning for each source table space and index space without a corresponding target table space or index space. See line #10 in the source job example and the associated note 10.
 - Specify COPY parm DEFVCAT to provide Db2 Cloning Tool Table Space Cloning with the space data set high level qualifier for all the table spaces and index spaces that do not exist in the target catalog. Note that only one can be supplied. If the target data sets have different high level qualifiers, run multiple Db2 Cloning Tool Table Space Cloning source jobs, each with the correct DEFVCAT value. See line #10 in the source job example and the associated note 10.

Results



CAUTION: If COPY command keyword FUZZY-COPY(Y) is specified, then Db2 Cloning Tool Table Space Cloning will not stop the source table spaces and index spaces. However, we *do not recommend* this, because if the table spaces and index spaces are in RW status, there are data integrity issues.

Source job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
/*JOBPARM S=srcsys
/**
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                            *
/*****
/**
/** Scenario 2 - one or more source spaces do not exist on the target
/**              and CKZ controls the copy process
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** srcsys  - name of source system
/** hlq1    - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2    - SYSMDUMP high level qualifier
/** dsnvrl  - DB2 high level qualifier
/** parmbr  - CKZ PARMLIB member
/** dumpsc  - SYSMDUMP STORCLAS if needed
/** dumpmc  - SYSMDUMP MGMTCLAS if needed
/** hlq3    - CKZ users high level qualifier
/** lstdmbr - LISTDEF member name
/** trss    - target DB2 subsystem
/** dddd    - DD DISP value if needed (must enable)
/** uuuu    - DD UNIT value if needed (must enable)
/** vvvvvv  - DD VOLUME value if needed (must enable)
/** ssss    - source DB2 subsystem
/** defsqlid - default SQLID applied to LISTDEF objects if needed
/** ipaddr  - target IP if using TCPIP (delete if not needed)
/** hlqtrgob - VCAT for target objects that do not exist
/** ocs     - object creator on source DB2 (delete if not needed)
/** oct     - object creator on target DB2 (delete if not needed)
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
/**
/**
/** CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD SYSOUT=*
/**
/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG DD SYSOUT=*
/**
/**
/** dump DDs
/**
/**SYSDUMP DD SYSOUT=* Note: SYSMDUMP preferred
/**SYSMDUMP DD DSN=hlq2.SYSMDUMP,DISP=(NEW,CATLG),
// STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/** SPACE=(CYL,(50,30),RLSE),
/** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**ABNLIGNR DD DUMMY do not remove if using ABENDAID
/**
/**
/** CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
/** to select the source spaces to be copied to the target.
/**
6 //CKZLSTDF DD DISP=SHR,DSN=hlq3.LISTDEF(lstdmbr)
/**
/**
/** CKZSYNC - output commands that will be used by the target
/** job to make the VSAM object(s) accessible on the target
/** DB2 subsystem.

```

```

7  //*
  //CKZSYNC DD DISP=OLD,DSN=hlq3.SYNCDDB2(1stdmbr)
  //*
  //*
  //* CKZIDCAM - output IDCAMS deletes and renames for all data sets
  //* copied to F0001, rather than I0001 or J0001.
  //*
  //CKZIDCAM DD DISP=OLD,DSN=hlq3.IDCAMS(1stdmbr)
  //*
  //*
  //* CKZSTPT - output DB2 stop commands for all target spaces
  //*
  //CKZSTPT DD DISP=OLD,DSN=hlq3.CMDSSTPT(1stdmbr)
  //*
  //*
  //* DSS output DD if required, see HLQDDDF
  //*
  //MYOUTDD DD DISP=dddd,UNIT=uuuu,VOL=SER=(vvvvvv)
  //*
  //*****
  //* sample control statements
  //*****
  //*
  //CKZIN DD *
  /*
  /*
  /* Source DB2 subsystem name and default SQLID if needed
  /*
  8 SET LOCAL-SSID(ssss) DEFAULT-SQLID(defsqlid)
  /*
  /*
  /* CKZ sends this DD name to ADRDSSU as output (enable if needed)
  /*
  9 /* HLQDDDF HLQNAME(trghlq) DDNAME(MYOUTDD)
  /*
  /*
  /*
  /* The COPY command invokes DFSMSdss program, ADRDSSU, to
  /* execute FlashCopy or SnapShot and prepares the object ID
  /* translation parameters for the target job. Or, if COPY
  /* keyword DATA-MOVER(PGM(NONE)) is present, CKZ outputs
  /* commands to assist the manual copy process.
  /*
  10 COPY TARGET-DB2(SSID(trss) -
      IP(ipaddr) -
      DEFVCAT(hlqtrgob) ) -
      SYNCDDB2-DDN(CKZSYNC) -
      IDCAMS-DDN(CKZIDCAM) -
      STOP-TARGET-DDN(CKZSTPT) -
      SIM(N) -
      DATA-MOVER( PGM(ADRDSSU)
      FASTREP(PREF)) -
      COPY-IF-NO-DB2-TARGET-OBJECTS(Y) -
      OBJECT-TRANSLATE(CREATOR,ocs,oct)-
      AUTO-STOP-TARGET-SPACE(Y) -
      AUTO-START-TARGET-SPACE(Y) -
      FUZZY-COPY(N) -
      RESET-LOGRBA(Y)

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated Db2 table spaces and index spaces, Db2 start and stop space command status and DFSMSdss program ADRDSSU commands and status. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG - Displays LISTDEF processing and Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. CKZLSTDF - Db2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control

statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *

7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target Db2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. The SET LOCAL command specifies the local Db2 subsystem for the source job and the TCP/IP server job.
9. The optional HLQDDDF command may be used to pass input and/or output DDs to ADRDSSU. This may be useful to pass volsers to ADRDSSU for non-SMS output volume allocations.
10. The COPY command invokes the DFSMSdss program ADRDSSU to execute FlashCopy or SnapShot and prepares the object ID translation parameters for the target job.

Step 3 (Optional): Create missing target table spaces and index spaces

As the target table spaces and index spaces did not exist when the Db2 Cloning Tool Table Space Cloning source in step 2 was run, there are no STOP commands for the newly created table spaces and index spaces. If needed, at this point another Db2 Cloning Tool Table Space Cloning source job can be run to output these stop commands.

Use PGM(NONE) and no stops will be issued to the existing source spaces. Additionally, delete the IDCAMS-DDN parameter from the source job. If this delete is not performed, the correct IDCAMS-DDN data set will be overwritten. For all missing target table spaces and index spaces, the next step will delete these data sets and rename the copied data sets from F0001 to I0001.

Step 4: Set up and execute optional FIX job (CKZFIX)

The purpose of the FIX job is to facilitate making the data sets copied to the target subsystem available to the target Db2.

When Db2 Cloning Tool Table Space Cloning detects missing target Db2 table spaces and index spaces, parameters are written to data sets which will stop the missing table spaces and index spaces (after the user creates them), delete the new VSAM objects and rename the VSAM objects copied by Db2 Cloning Tool Table Space Cloning.

Sample JCL for the FIX job is in the product JCL library, member CKZFIX. Set up the job to point to the same data sets as STOP-TARGET-DDN and IDCAMS-DDN.

The CKZFIX job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //jobcard
  /*JOBPARM S=trgsys
  /*
  /******
  /*          COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015          *
  /*          ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use when missing DB2 objects created after the copy.
  /**
  /** variables to be filled in ...
  /**
  /**   jobcard - job card
  /**   trgsys  - name of target system
  /**   dsnvrl - DB2 high level qualifier
  /**   hlq3   - CKZ users high level qualifier
  /**   lstdmbr - LISTDEF member name
  /**
  /**
  /******
  /** Stop the DB2 created table space(s) AND index(es) *
  /** for the deletion and rename step to follow.      *
  /******
  2 //STOP      EXEC PGM=IKJEFT01
  /*
```

```

//*
//STEPLIB DD DISP=SHR,DSN=dsnvrl.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//*
//* SYSTSIN DD was CKZSTPT DD in the source job CKZ2SRC
//*
3 //SYSTSIN DD DISP=SHR,DSN=h1q3.CMDSSTPT(1stdmbr)
4 //* DSN SYSTEM(trss)
//* -STOP DATABASE(DBASEA) SPACE(TSPACEA)
//* -STOP DATABASE(DBASEA) SPACE(INDEXA)
//*
//*
//*****
//* Delete the DB2 created VSAM data set(s) and *
//* rename the cloned data set(s) *
//*****
5 //DELRNAME EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//*
//*
//* SYSIN DD was CKZIDCAM DD in the source job CKZ2SRC
//*
6 //SYSIN DD DISP=SHR,DSN=h1q3.IDCAMS(1stdmbr)
//*
//* DELETE DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001 -
//* CLUSTER NOERASE PURGE
//* ALTER -
//* DSN081D.DSNDBC.DBASEA.TSPACEA.F0001.A001 -
//* NEWNAME(DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001)
//* ALTER -
//* DSN081D.DSNDBD.DBASEA.TSPACEA.F0001.A001 -
//* NEWNAME(DSN081D.DSNDBD.DBASEA.TSPACEA.I0001.A001)
//*

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN. This data set is only available if the source job was resubmitted after the missing target table spaces and index spaces have been created.
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.
5. Execution of IDCAMS – To delete the VSAM objects created by the target Db2 when the table spaces and index spaces were created and rename the VSAM objects copied from the source Db2.
6. SYSIN – Data set referenced by source job COPY command keyword IDCAMS-DDN.

Step 5: Set up for a re-run of the source job

Re-running the source job at this point will capture the object IDs of the newly created Db2 table spaces and index spaces and update the CKZSYNC and CKZSQL (if required) data sets used as input for the SYNCDB2 command.

The COPY command keyword must be DATA-MOVER(PGM(NONE)). With PGM(NONE), no Db2 stop or start space commands will be issued.

Step 6: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and SYNCDB2 command to make the VSAM objects accessible on the target subsystem. The target job assumes all target spaces are stopped. It will optionally start the target table spaces and index spaces.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example and associated note.

Procedure

1. Specify the PARMLIB data set. Note that the Db2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCPIP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target, and TCP/IP server jobs.

See line #3 in the target job example and the associated note 3.

2. Specify the input data set where the Db2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, on line #7.

See line #6 in the target job example and the associated note 6.

Target job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The target job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
//*JOBPARM S=trgsys
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                            *
/*****
/**
/** Scenarios 1-4 - the target job is the same for all 4 scenarios.
/** The only change is the data set specified in CKZSQL and CKZIN.
/** These are output from the source job.
/**
/** The target job must be run on the target DB2 system.
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** trgsys - name of target system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSMDUMP high level qualifier
/** dsnvrl - DB2 high level qualifier
/** parmbr - CKZ PARMLIB member
/** dumpsc - SYSMDUMP STORCLAS if needed
/** dumpmc - SYSMDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmbr - LISTDEF member name
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
```

```

3 //CKZINI DD DISP=SHR,DSN=h1q1.SCKZPARM(parmmb)
//*
//*
//* CKZPRINT - primary message output, required
//*
4 //CKZPRINT DD SYSOUT=*
//*
//*
//* CKZLOG - detailed message output, optional
//*
5 //CKZLOG DD SYSOUT=*
//*
//*
//* dump DDs
//*
//*SYSDUMP DD SYSOUT=* Note: SYSDUMP preferred
//*SYSDUMP DD DSN=h1q2.SYSDUMP,DISP=(NEW,CATLG),
//* STORCLAS=dumpsc,MGMTCLAS=dumpmc,
//* SPACE=(CYL,(50,30),RLSE),
//* DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
//*ABNLIGNR DD DUMMY do not remove if using ABENDAID
//*
//*
//* CKZIN - input commands that will be used by the target
//* job to make the VSAM object(s) accessible on the target
//* DB2 subsystem. Created by the source job.
//*
6 //CKZIN DD DISP=SHR,DSN=h1q3.SYNCDB2(1stdmbr)
//*
//*****
//* sample control statements as would appear in SYNCDB2 member
//* Note: all lines have /* added in column 1 for JCL compatibility.
//*****
//*
//*CKZIN DD *
//* 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
//* SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
//*
//*SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCAN0(N)
//*
//*
//* TABLE SPACES - COPIED SUCCESSFULLY */
//*
//*
//*SYNCDB2 TARGET-SSID (V81S) -
//* TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
//* SPACE-TYPE (TS ) -
//* PAGE-SIZE (4) -
//* VCAT (DSN081D) -
//* STOGROUP (ZSG884) -
//* START-SPACE (Y) -
//* RESET-LOGRBA (Y) -
//* XLATE (DBID,X'0120,X'0113, -
//* PSID,X'0002,X'0047, -
//* TSOB,X'0001,X'0046, -
//* TBOB,X'0003,X'0048, -
//* TBOB,X'0008,X'004D, -
//* TBOB,X'000D,X'0052)
//*
//*SYNCDB2 TARGET-SSID (V81S) -
//* TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
//* SPACE-TYPE (TS ) -
//* PAGE-SIZE (4) -
//* VCAT (DSN081D) -
//* STOGROUP (ZSG884) -
//* START-SPACE (Y) -
//* RESET-LOGRBA (Y) -
//* XLATE (DBID,X'0120,X'0113, -
//* PSID,X'0024,X'0013, -
//* TSOB,X'0023,X'0012, -
//* TBOB,X'0025,X'0014, -
//* TBOB,X'002A,X'0019, -
//* TBOB,X'002F,X'001E)
//*
//*SYNCDB2 TARGET-SSID (V81S) -
//* TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
//* SPACE-TYPE (TP ) -
//* PAGE-SIZE (4) -
//* VCAT (DSN081D) -
//* STOGROUP (ZSG884) -
//* START-SPACE (Y) -
//* RESET-LOGRBA (Y) -

```

```

//* XLATE (DBID,X'0120,X'0113, -
//* PSID,X'0046,X'0035, -
//* TSOB,X'0045,X'0034, -
//* TBOB,X'0047,X'0036)
//*
//*SYNCDB2 TARGET-SSID (V81S) -
//* TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
//* SPACE-TYPE (TP ) -
//* PAGE-SIZE (4) -
//* VCAT (DSN081D) -
//* STOGROUP (ZSG884) -
//* START-SPACE (Y) -
//* RESET-LOGRBA (Y) -
//* XLATE (DBID,X'0120,X'0113, -
//* PSID,X'0046,X'0035, -
//* TSOB,X'0045,X'0034, -
//* TBOB,X'0041,X'0036)

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, CKZIN control parameters, Db2 SQL execution status, and SYNCDB2 status and START Db2 command status for each data set processed. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG displays the Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/ results of the commands and detailed information about each Db2 page access. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Chapter 15. Set up procedures for copy by data set for all other methodologies

This topic contains two Db2 Cloning Tool Table Space Cloning setup procedures for you to use if data set copies are to be created using copy methodologies other than FlashCopy or SnapShot via DFSMSdss or TimeFinder/Clone. Choose which procedure to use based on the following scenarios.

Setup procedures for scenarios using FlashCopy, SnapShot, or TimeFinder/Clone are documented in Chapter 14, “Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone,” on page 219.

Setup for scenario 3: Other copy methodologies - all target table spaces and index spaces already exist in Db2 catalog

Use the procedure in the following table for all other copy methodologies, other than FlashCopy, SnapShot, or TimeFinder/Clone, to copy source table spaces and index spaces to the target Db2 subsystem when the target table spaces and index spaces already exist in the Db2 catalog.

| <i>Table 49. Db2 Cloning Tool Table Space Cloning - Setting up for other copy methodologies</i> |
|--|
| Setup steps for other methodologies |
| “Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog” on page 237. |
| “Step 2: Set up and execute source job (CKZ3SRC)” on page 238. |
| “Step 3 (Optional): Set up and execute stop target job (CKZSTPT)” on page 241. |
| “Step 4 (Optional): Set up and execute stop source job (CKZSTPS)” on page 242. |
| “Step 5: Set up and execute non-Db2 Cloning Tool Table Space Cloning job to execute data set copies outside Db2 Cloning Tool Table Space Cloning” on page 243. |
| “Step 6 (Optional): Set up and execute job to re-start source Db2 table spaces and index spaces (CKZSTRS)” on page 243. |
| “Step 7: Set up and execute target job (CKZTRG)” on page 244. |

Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog

The first step when copying by data set using other methodologies to verify/create the target objects in the target Db2 catalog.

Db2 associates Db2 table spaces and index spaces with specific VSAM data set names. For example:

```
VCAT.DSNDBD.DBTEST1.TESTTBL.I0001.A001
```

where:

- Node 1: VCAT - the data set high level qualifier
- Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion
- Node 3: DBTEST1 - the database name in the Db2 catalog
- Node 4: TESTTBL - the table space or index space name in the Db2 catalog
- Node 5: Either I0001 or J0001 (a switch indicator)
- Node 6: A001 to E094, a partition number or Db2 extent number

The correctly named Db2 table spaces and index spaces must exist in the target Db2 catalog for Db2 to recognize and use the VSAM data sets cloned to the target by Db2 Cloning Tool Table Space Cloning.

When Db2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets cloned from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. Db2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target Db2.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: Db2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up and execute source job (CKZ3SRC)

The source job is comprised of several commands and phases to control the selection of the Db2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

Before you begin

Note: It is recommended that you first run the source job with CHECK-DATASET-COMPATIBILITY(Y) in simulation mode (PGM(NONE), or SIM(A) if PGM(ADRDSSU) is specified) to determine whether the VSAM attributes of the source data sets and the target data sets are compatible. After you resolve any data set incompatibilities, then set CHECK-DATASET-COMPATIBILITY to N and run the source job with PGM and SIM set as desired. To avoid unnecessary CPU and I/O time, do not use CHECK-DATASET-COMPATIBILITY(Y) in a non-simulation run of the source job.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ3SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to Db2 Cloning Tool Table Space Cloning using the Db2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where Db2 Cloning Tool Table Space Cloning will build the input parameters for the target job.

See line #7 in the source job example and the associated note 7.

3. Identify the source Db2 subsystem in the SET command, keyword LOCAL-SSID(*n*), where *n* is the name of your source Db2 subsystem.

See line #9 in the source job example and the associated note 9.

4. Identify the target Db2 subsystem information and copy method in the COPY command.

- Identify the target Db2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(*n*)) where *n* is the name of your target Db2 subsystem. See line #10 in the source job example and the associated note 10.
- Specify keyword DATA-MOVER(PGM(NONE)) in the COPY command to indicate that the copy is to be done outside of Db2 Cloning Tool Table Space Cloning. See line #10 in the source job example and the associated note 10.

Source job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=srcsys
/**
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                          *
/*****
/**
/** Scenario 3 - all spaces to be copied exist on the target and
/**              copies are done outside of CKZ
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** srcsys - name of source system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSMDUMP high level qualifier
/** dsnvrl - DB2 high level qualifier
/** parmmb - CKZ PARMLIB member
/** dumpsc - SYSMDUMP STORCLAS if needed
/** dumpmc - SYSMDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmb - LISTDEF member name
/** trss - target DB2 subsystem
/** dddd - DD DISP value if needed (must enable)
/** uuuu - DD UNIT value if needed (must enable)
/** vvvvvv - DD VOLUME value if needed (must enable)
/** ssss - source DB2 subsystem
/** defsqlid - default SQLID applied to LISTDEF objects if needed
/** ipaddr - target IP if using TCPIP (delete if not needed)
/** ocs - object creator on source DB2 (delete if not needed)
/** oct - object creator on target DB2 (delete if not needed)
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmmb)
/**
/**
/** CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD SYSOUT=*
/**
```

```

/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG DD SYSOUT=*
/**
/** dump DDs
/**
/**SYSDUMP DD SYSOUT=* Note: SYSDUMP preferred
/**SYSDUMP DD DSN=h1q2.SYSDUMP,DISP=(NEW,CATLG),
/** STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/** SPACE=(CYL,(50,30),RLSE),
/** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**ABNLIGNR DD DUMMY do not remove if using ABENDAID
/**
/**
/** CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
/** to select the source spaces to be copied to the target.
/**
6 //CKZLSTDF DD DISP=SHR,DSN=h1q3.LISTDEF(1stdmbr)
/**
/**
/** CKZSYNC - output commands that will be used by the target
/** job to make the VSAM object(s) accessible on the target
/** DB2 subsystem.
/**
7 //CKZSYNC DD DISP=OLD,DSN=h1q3.SYNCDB2(1stdmbr)
/**
/**
/** CKZCOPY - output list to copy with from data set name and to
/** data set name
/**
8 //CKZCOPY DD DISP=OLD,DSN=h1q3.COPYDSNS(1stdmbr)
/**
/**
/** CKZSTPT - output DB2 stop commands for all target spaces
/**
//CKZSTPT DD DISP=OLD,DSN=h1q3.CMDSSTPT(1stdmbr)
/**
/**
/** CKZSTPS - output DB2 stop commands for all source spaces
/**
//CKZSTPS DD DISP=OLD,DSN=h1q3.CMDSSTPS(1stdmbr)
/**
/**
/** CKZSTRS - output DB2 stop commands for all source spaces
/**
//CKZSTRS DD DISP=OLD,DSN=h1q3.CMDSSTRS(1stdmbr)
/**
/**
/**
/** sample control statements
/**
//CKZIN DD *
/* */
/* */
/* Source DB2 subsystem name and default SQLID if needed */
/* */
9 SET LOCAL-SSID(ssid) DEFAULT-SQLID(defsqlid)
/* */
/* */
/* The COPY command prepares the objects ID translation */
/* parameters for the target job and outputs commands to */
/* assist the manual copy process. */
/* */
10 COPY TARGET-DB2(SSID(trss) -
IP(ipaddr) -
DATASETS-TO-COPY-DDN(CKZCOPY) -
SYNCDB2-DDN(CKZSYNC) -
STOP-TARGET-DDN(CKZSTPT) -
STOP-SOURCE-DDN(CKZSTPS) -
START-SOURCE-DDN(CKZSTRS) -
SIM(N) -
DATA-MOVER( PGM(NONE)) -
COPY-IF-NO-DB2-TARGET-OBJECTS(N) -
OBJECT-TRANSLATE(CREATOR,ocs,oct)-
AUTO-STOP-TARGET-SPACE(Y) -
AUTO-START-TARGET-SPACE(Y) -
RESET-LOGRBA(Y)

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated Db2 table spaces and index spaces, Db2 start and stop space command status and DFSMSdss program ADRDSSU commands and status. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG - Displays LISTDEF processing and Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. CKZLSTDF - Db2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *.
7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target Db2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. DD for CKZCOPY - The CKZCOPY data sets contains a list of input and output data sets if the keyword for the COPY command is: DATA-MOVER(PGM(NONE)). The purpose of this listing is to assist copying the selected VSAM objects outside of Db2 Cloning Tool Table Space Cloning.
9. The SET LOCAL command specifies the local Db2 subsystem for the source job and the TCP/IP server job.
10. COPY Command - The COPY command with DATA-MOVER(PGM(NONE)) assumes the copy of the data sets is to be done outside of Db2 Cloning Tool Table Space Cloning, and only prepares the object ID transaction parameters for the target job and outputs Db2 start and stop space commands and IDCAMS delete/define commands into data sets if requested.

Step 3 (Optional): Set up and execute stop target job (CKZSTPT)

The purpose of the CKZSTPT job is to facilitate copying data sets outside of Db2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), Db2 Cloning Tool Table Space Cloning writes Db2 stop parameters for the target Db2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-TARGET-DDN(?).

Sample JCL for the CKZSTPT job is in the product JCL library, member CKZSTPT. Set up the job to point to the same data set as STOP-TARGET-DDN.

The CKZSTPT job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //jobcard
  /*JOBPARM S=trgsys
  /******
  /*          COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015          *
  /*          ALL RIGHTS RESERVED                                *
  /******
  /*
  /* Use to stop target objects if a copy method outside CKZ
  /* is used to copy data sets from the source to the target.
  /*
  /* variables to be filled in ...
  /*
  /* jobcard - job card
  /* trgsys - name of target system
  /* dsnvrl - DB2 high level qualifier
  /* hlq3 - CKZ users high level qualifier
  /* lstdmbr - LISTDEF member name
  /*
```

```

//*
//*****
//* Stop the DB2 target table and index spaces *
//*****
//*
2 //STOPT EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=dsnvrl.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//*
//* SYSTSIN DD was CKZSTPT DD in the source job.
//*
3 //SYSTSIN DD DISP=SHR,DSN=h1q3.CMDSSTPT(1stdmbr)
4 //* DSN SYSTEM(trss)
//* -STOP DATABASE(DBASEA) SPACE(TSPACEA)
//* -STOP DATABASE(DBASEA) SPACE(INDEXA)
//*

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.

Step 4 (Optional): Set up and execute stop source job (CKZSTPS)

The purpose of the CKZSTPS job is to facilitate copying data sets outside of Db2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), Db2 Cloning Tool Table Space Cloning writes Db2 stop parameters for the source Db2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-SOURCE-DDN(ddname).

Sample JCL for the CKZSTPTS job is in the product JCL library, member CKZSTPTS. Set up the job to point to the same data set as STOP-SOURCE-DDN.

The CKZSTPTS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=srcsys
//*****
//* COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015 *
//* ALL RIGHTS RESERVED *
//*****
//*
//* Use to stop source spaces if a copy method outside CKZ
//* is used to copy data sets from the source to the target.
//*
//* variables to be filled in ...
//*
//* jobcard - job card
//* srcsys - name of source system
//* dsnvrl - DB2 high level qualifier
//* h1q3 - CKZ users high level qualifier
//* 1stdmbr - LISTDEF member name
//*
//*
2 //STOPS EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=dsnvrl.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//*
//* SYSTSIN DD was CKZSTPS DD in the source job.
//*
3 //SYSTSIN DD DISP=SHR,DSN=h1q3.CMDSSTPS(1stdmbr)
4 //* DSN SYSTEM(srcss)
//* -STOP DATABASE(DBASEA) SPACE(TSPACEA)

```

```
//* -STOP DATABASE(DBASEA) SPACE(INDEXA)
//*
```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-SOURCE-DDN
4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.

Step 5: Set up and execute non-Db2 Cloning Tool Table Space Cloning job to execute data set copies outside Db2 Cloning Tool Table Space Cloning

Employ whatever procedures and syntax appropriate to your chosen replication tool.

If requested by the COPY parameter DATASETS-TO-COPY-DDN, Db2 Cloning Tool Table Space Cloning writes out a list of the source and target VSAM object pairs. The target data set names will have the correct high level qualifier and the correct switch (I/J) indicator. In addition, only the qualifying data set pairs will be listed.

A sample output:

```
* 06330 11:56:36.79 JOBNAME=JMX8SRC JOBID=J0044593
* SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S
*
*
* TABLE SPACE DATA SETS TO BE COPIED
*
* SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
* TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
*
* SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN2TSSC.I0001.A001
* TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN2TSSC.J0001.A001
*
* SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001
* TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.J0001.A001
```

Step 6 (Optional): Set up and execute job to re-start source Db2 table spaces and index spaces (CKZSTRS)

The purpose of the CKZSTRS job is to facilitate copying data sets outside of Db2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), Db2 Cloning Tool Table Space Cloning writes Db2 start parameters for the source Db2 table spaces and index spaces to a data set pointed to by COPY keyword START-SOURCE-DDN(?).

Sample JCL for the CKZSTRS job is in the product JCL library, member CKZSTRS. Set up the job to point to the same data set as START-SOURCE-DDN.

The CKZSTRS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //jobcard
/*JOBPARM S=srcsys
//*****
/* COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015 *
/* ALL RIGHTS RESERVED *
//*****
/*
/* Use to start source objects if a copy method outside CKZ
/* is used to copy data sets from the source to the target.
/*
/* variables to be filled in ...
/*
/* jobcard - job card
```

```

 2  /**      srcsys   - name of source system
 3  /**      dsnvr1  - DB2 high level qualifier
 4  /**      hlq3    - CKZ users high level qualifier
 5  /**      lstdmbr - LISTDEF member name
 6  /**
 7  /**
 8  2  //STARTS   EXEC   PGM=IKJEFT01
 9  //STEPLIB  DD     DISP=SHR,DSN=dsnvr1.SDSNLOAD
10  //SYSPRINT DD     SYSOUT=*
11  //SYSTSPRT DD     SYSOUT=*
12  /**
13  /**
14  /**      SYSTSIN DD was CKZSTRS DD in the source job.
15  /**
16  3  //SYSTSIN  DD     DISP=SHR,DSN=hlq3.CMDSSTRS(lstdmbr)
17  4  /**      DSN     SYSTEM(srss)
18  /**      -START  DATABASE(DBASEA) SPACE(TSPACEA)
19  /**      -START  DATABASE(DBASEA) SPACE(INDEXA)
20  /**

```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword START-SOURCE-DDN
4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.

Step 7: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and SYNCDDB2 command to make the VSAM objects accessible on the target subsystem. It will optionally stop and start the target table spaces and index spaces.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. A detailed example follows.

Procedure

1. Specify the PARMLIB data set.
Note that the Db2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCP/IP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target and TCP/IP server jobs. See line #3 in the target job example and the associated note 3.
2. Specify the input data set where the Db2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, line 7.
See line #6 in the target job example and the associated note 6.

Target job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

 1  //jobcard
 2  /*JOBPARM S=trgsys
 3  /*******
 4  /**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
 5  /**      ALL RIGHTS RESERVED                            *
 6  /*******

```

```

/**
/** Scenarios 1-4 - the target job is the same for all 4 scenarios.
/** The only change is the data set specified in CKZSQL and CKZIN.
/** These are output from the source job.
/**
/** The target job must be run on the target DB2 system.
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** trgsys - name of target system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSDUMP high level qualifier
/** dsnvrl - DB2 high level qualifier
/** parmmbr - CKZ PARMLIB member
/** dumpsc - SYSDUMP STORCLAS if needed
/** dumpmc - SYSDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmbr - LISTDEF member name
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmmbr)
/**
/**
/** CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD SYSOUT=*
/**
/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG DD SYSOUT=*
/**
/**
/** dump DDs
/**
/**SYSDUMP DD SYSOUT=* Note: SYSDUMP preferred
/**SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
/** STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/** SPACE=(CYL,(50,30),RLSE),
/** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**ABNLIGNR DD DUMMY do not remove if using ABENDAID
/**
/**
/** CKZIN - input commands that will be used by the target
/** job to make the VSAM object(s) accessible on the target
/** DB2 subsystem. Created by the source job.
/**
6 //CKZIN DD DISP=SHR,DSN=hlq3.SYNCDB2(lstdmbr)
/**
/**
/** sample control statements as would appear in SYNCDB2 member
/** Note: all lines have /* added in column 1 for JCL compatibility.
/**
/**
/**CKZIN DD *
/** 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
/** SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
/**
/**SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCAN0(N)
/**
/**
/** TABLE SPACES - COPIED SUCCESSFULLY */
/**
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -

```

```

/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0002,X'0047, -
/** TSOB,X'0001,X'0046, -
/** TBOB,X'0003,X'0048, -
/** TBOB,X'0008,X'004D, -
/** TBOB,X'000D,X'0052)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0024,X'0013, -
/** TSOB,X'0023,X'0012, -
/** TBOB,X'0025,X'0014, -
/** TBOB,X'002A,X'0019, -
/** TBOB,X'002F,X'001E)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0046,X'0035, -
/** TSOB,X'0045,X'0034, -
/** TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0046,X'0035, -
/** TSOB,X'0045,X'0034, -
/** TBOB,X'0047,X'0036)

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT - Displays CKZINI tokens, CKZIN control parameters, Db2 SQL execution status, and SYNCDB2 status and START Db2 command status for each data set processed. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG - Displays the Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each Db2 page access. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Setup for scenario 4: Other copy methodologies - some or all target table spaces and index spaces do not exist in Db2 catalog

Ideally, all target Db2 table spaces and index spaces will exist before the Db2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy and there

is not enough time to create the target table spaces and index spaces within the window, then the following procedure is available.

You may also use this procedure (shown in the following table) for the table spaces and index spaces that are inadvertently missed. The Db2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. The output to some data sets will vary depending on whether the table spaces and index spaces are missing on the target or not..

| Setup steps for other copy methodologies when some or all target objects do not exist |
|--|
| “Step 1: Create target Db2 table spaces and index spaces in target Db2 catalog” on page 247. |
| “Step 2: Set up source job (CKZ4SRC)” on page 248. |
| “Step 3 (Optional): Set up stop target job (CKZSTPT)” on page 252. |
| “Step 4 (Optional): Set up and execute stop source job (CKZSTPS)” on page 242. |
| “Step 5: Set up non-Db2 Cloning Tool Table Space Cloning job to execute data set copies outside Db2 Cloning Tool Table Space Cloning” on page 253. Use the list of source and target data sets provided by Db2 Cloning Tool Table Space Cloning. |
| “Step 6 (Optional): Set up and execute job to re-start source Db2 table spaces and index spaces (CKZSTRS)” on page 243. |
| “Step 7 (Optional): Create missing target table spaces and index spaces” on page 255 the Db2 Cloning Tool Table Space Cloning source job in Step 2 of this procedure found missing from the target. |
| “Step 8 (Optional): Set up job to make copied VSAM objects available to target Db2 (CKZFIX)” on page 255. |
| “Step 9: Set up for a re-run of the source job” on page 256. Re-running the source job will update the control parameters for the target job so that it can work on the newly created table spaces and index spaces. |
| “Step 10: Set up and execute target job (CKZTRG)” on page 256. |

Step 1: Create target Db2 table spaces and index spaces in target Db2 catalog

When some or all target table spaces and index spaces do not exist, the first step when copying by data set using copy methodologies other than FlashCopy, SnapShot, or TimeFinder/Clone is to create the target objects in the target Db2 catalog.

Db2 associates Db2 table spaces and index spaces with specific VSAM data set names. For example:

```
VCAT.DSNDBD.DBTEST1.TESTTBL.I0001.A001
```

where:

- Node 1: VCAT - the data set high level qualifier
- Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion
- Node 3: DBTEST1 - the database name in the Db2 catalog
- Node 4: TESTTBL - the table space or index space name in the Db2 catalog
- Node 5: Either I0001 or J0001 - a switch indicator
- Node 6: A001 to E094A - a partition number or Db2 extent number

The correctly named Db2 table spaces and index spaces must exist in the target Db2 catalog for Db2 to recognize and use the VSAM data sets cloned to the target by Db2 Cloning Tool Table Space Cloning.

When Db2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets copied from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. Db2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target Db2. Therefore you must create those target table spaces and index spaces if they do not exist.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: Db2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up source job (CKZ4SRC)

The source job is comprised of several commands and phases to control the selection of the Db2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

Before you begin

Note: It is recommended that you first run the source job with CHECK-DATASET-COMPATIBILITY(Y) in simulation mode (PGM(NONE), or SIM(A) if PGM(ADRDU) is specified) to determine whether the VSAM attributes of the source data sets and the target data sets are compatible. After you resolve any data set incompatibilities, then set CHECK-DATASET-COMPATIBILITY to N and run the source job with PGM and SIM set as desired. To avoid unnecessary CPU and I/O time, do not use CHECK-DATASET-COMPATIBILITY(Y) in a non-simulation run of the source job.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ4SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to Db2 Cloning Tool Table Space Cloning using the Db2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where Db2 Cloning Tool Table Space Cloning will build the input parameters for the target job.

See line #7 in the source job example and the associated note 7.

3. Identify the target Db2 subsystem information and copy method in the COPY command.

- Identify the target Db2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(*n*)) where *n* is the name of your target Db2 subsystem. See line #13 in the source job example and the associated note 13.
- Identify the default VCAT and target Db2 subsystem to CKZIN in the COPY command, keyword DEFVCAT(*n*) where *n* is a high level qualifier for the target Db2 subsystem data sets used by the target job if one or more target table spaces and index spaces do not exist in the Db2 catalog. See line #13 in the source job example and the associated note 13.
- Since using a copy methodology other than FlashCopy, SnapShot, or TimeFinder/Clone to copy table spaces and index spaces, specify keyword DATA-MOVER(PGM(NONE)) in the COPY command to indicate that the copy is to be done outside of Db2 Cloning Tool Table Space Cloning. See line #13 in the source job example and the associated note 13.

Source job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=srcsys
/**
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                            *
/*****
/**
/** Scenario 4 - one or more source spaces do not exist on the target
/**              and copies are done outside of CKZ
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** srcsys - name of source system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSMDUMP high level qualifier
/** dsnvrl - DB2 high level qualifier
/** parmbr - CKZ PARMLIB member
/** dumpsc - SYSMDUMP STORCLAS if needed
/** dumpmc - SYSMDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmb - LISTDEF member name
/** trss - target DB2 subsystem
/** dddd - DD DISP value if needed (must enable)
/** uuuu - DD UNIT value if needed (must enable)
/** vvvvvv - DD VOLUME value if needed (must enable)
/** ssss - source DB2 subsystem
/** defsqlid - default SQLID applied to LISTDEF objects if needed
/** ipaddr - target IP if using TCPIP (delete if not needed)
/** hlqtrgob - VCAT for target objects that do not exist
/** ocs - object creator on source DB2 (delete if not needed)
/** oct - object creator on target DB2 (delete if not needed)
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
/**
/**
/** CKZPRINT - primary message output, required
/**
```

```

4 //CKZPRINT DD SYSOUT=*
  /**
  /**
  /** CKZLOG - detailed message output, optional
  /**
5 //CKZLOG DD SYSOUT=*
  /**
  /**
  /** dump DDs
  /**
  /***SYSUDUMP DD SYSOUT=* Note: SYSDUMP preferred
  /***SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
  /** STORCLAS=dumpsc,MGMTCLAS=dumpmc,
  /** SPACE=(CYL,(50,30),RLSE),
  /** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
  /***ABNLIGNR DD DUMMY do not remove if using ABENDAID
  /**
  /**
  /*** CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
  /*** to select the source spaces to be copied to the target.
  /**
6 //CKZLSTDF DD DISP=SHR,DSN=hlq3.LISTDEF(lstdmbr)
  /**
  /**
  /*** CKZSYNC - output commands that will be used by the target
  /*** job to make the VSAM object(s) accessible on the target
  /*** DB2 subsystem.
  /**
7 //CKZSYNC DD DISP=OLD,DSN=hlq3.SYNCDB2(lstdmbr)
  /**
  /**
  /*** CKZCOPY - output list to copy with from data set name and to
  /*** data set name
  /**
8 //CKZCOPY DD DISP=OLD,DSN=hlq3.COPYDNS(lstdmbr)
  /**
  /**
  /*** CKZIDCAM - output IDCAMS deletes and renames for all data sets
  /*** copied to F0001, rather than I0001 or J0001.
  /**
  /***CKZIDCAM DD DISP=OLD,DSN=hlq3.IDCAMS(lstdmbr)
  /**
  /**
  /*** CKZSTPT - output DB2 stop commands for all target spaces
  /**
9 //CKZSTPT DD DISP=OLD,DSN=hlq3.CMDSSTPT(lstdmbr)
  /**
  /**
  /*** CKZSTPS - output DB2 stop commands for all source spaces
  /**
10 //CKZSTPS DD DISP=OLD,DSN=hlq3.CMDSSTPS(lstdmbr)
  /**
  /**
  /*** CKZSTRS - output DB2 stop commands for all source spaces
  /**
11 //CKZSTRS DD DISP=OLD,DSN=hlq3.CMDSSTRS(lstdmbr)
  /**
  /**
  /*** DSS output DD if required, see HLQDDDF below
  /**
  /***MYOUTDD DD DISP=dddd,UNIT=uuuu,VOL=SER=(vvvvvv)
  /**
  /*******
  /*** sample control statements
  /*******
  /**
  /**CKZIN DD *
  /**
  /**
  /** Source DB2 subsystem name and default SQLID if needed
  /**
  /**
12 SET LOCAL-SSID(ssid) DEFAULT-SQLID(defsqlid)
  /**
  /**
  /** CKZ sends this DD name to ADRDSSU as output (enable if needed)
  /**
  /**
  /** HLQDDDF HLQNAME(trighlq) DDNAME(MYOUTDD)
  /**
  /**
  /**
  /** The COPY command prepares the objects ID translation
  /** parameters for the target job and outputs commands to
  /** assist the manual copy process.
  /**

```

```

13  /*                                     */
    COPY TARGET-DB2(SSID(trss)           -
        IP(ipaddr)                       -
        DEFVCAT(hlqtrgob) )              -
        DATASETS-T0-COPY-DDN(CKZCOPY)    -
        SYNCDB2-DDN(CKZSYNC)             -
        IDCAMS-DDN(CKZIDCAM)             -
        STOP-TARGET-DDN(CKZSTPT)         -
        STOP-SOURCE-DDN(CKZSTPS)        -
        START-SOURCE-DDN(CKZSTRS)       -
        SIM(N)                            -
        DATA-MOVER( PGM(NONE) )         -
        COPY-IF-NO-DB2-TARGET-OBJECTS(Y) -
        OBJECT-TRANSLATE(CREATOR,ocs,oct)-
        AUTO-STOP-TARGET-SPACE(Y)       -
        AUTO-START-TARGET-SPACE(Y)      -
        RESET-LOGRBA(Y)                  -

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated Db2 table spaces and index spaces, Db2 start and stop space command status and DFSMSdss program ADRDSSU commands and status. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG - Displays LISTDEF processing and Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. CKZLSTDF - Db2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *
7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target Db2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. DD for CKZCOPY - The CKZCOPY data sets contains a list of input and output data sets if the keyword for the COPY command is: DATA-MOVER(PGM(NONE)). The purpose of this listing is to assist in copying the selected VSAM objects outside of Db2 Cloning Tool Table Space Cloning
9. DD for CMDSSSTPT - The CMDSSSTPT data set contains the parameters that will be used by the CKZSTPT job to stop the target table spaces and index spaces if the copy is done outside of Db2 Cloning Tool Table Space Cloning.
10. DD for CMDSSSTPS - The CMDSSSTPS data set contains the parameters that will be used by the CKZSTPS job to stop the source table spaces and index spaces if the copy is done outside of Db2 Cloning Tool Table Space Cloning.
11. DD for CMDSSSTRS - The CMDSSSTRS data set contains the parameters that will be used by the CKZSTRS job to start the source table spaces and index spaces if the copy is done outside of Db2 Cloning Tool Table Space Cloning.
12. The SET LOCAL command specifies the local Db2 subsystem for the source job and the TCP/IP server job.
13. COPY Command - The COPY command with DATA-MOVER(PGM(NONE)) assumes the copy of the data sets was done outside of Db2 Cloning Tool Table Space Cloning, and only prepares the object ID transaction parameters for the target job and outputs Db2 start and stop space commands and IDCAMS delete/define commands into data sets if requested.

Step 3 (Optional): Set up stop target job (CKZSTPT)

The CKZSTPT job will help copy data sets outside of Db2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), Db2 Cloning Tool Table Space Cloning writes Db2 stop parameters for the existing target Db2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-TARGET-DDN(?). Note that Db2 Cloning Tool Table Space Cloning can only generate STOP commands for existing table spaces and index spaces. Another CKZ4SRC may be submitted after the target table spaces and index spaces are created to output all the target STOP commands.

Sample JCL for the CKZSTPT job is in the product JCL library, member CKZSTPT. Set up the job to point to the same data set as STOP-TARGET-DDN.

The CKZSTPT job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //jobcard
  /*JOBPARM S=trgsys
  /******
  /*      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015          *
  /*      ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use to stop target objects if a copy method outside CKZ
  /** is used to copy data sets from the source to the target.
  /**
  /** variables to be filled in ...
  /**
  /** jobcard - job card
  /** trgsys - name of target system
  /** dsnvrl - DB2 high level qualifier
  /** hlq3 - CKZ users high level qualifier
  /** lstdmbr - LISTDEF member name
  /**
  /**
  /******
  /** Stop the DB2 target table and index spaces          *
  /******
2 //STOPT EXEC PGM=IKJEFT01
  //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  //SYSPRINT DD SYSOUT=*
  //SYSTSPRT DD SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTPT DD in the source job.
  /**
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTPT(lstdmbr)
4 /** DSN SYSTEM(trss)
  /** -STOP DATABASE(DBASEA) SPACE(TSPACEA)
  /** -STOP DATABASE(DBASEA) SPACE(INDEXA)
  /**
```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.

Step 4 (Optional): Set up stop source job (CKZSTPS)

The purpose of the CKZSTPS job is to facilitate copying data sets outside of Db2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), Db2 Cloning Tool Table Space Cloning writes Db2 stop parameters for the source Db2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-SOURCE-DDN(ddname).

Sample JCL for the CKZSTPTS job is in the product JCL library, member CKZSTPTS. Set up the job to point to the same data set as STOP-SOURCE-DDN.

The CKZSTPTS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //jobcard
  /*JOBPARM S=srcsys
  /*******
  /**          COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015          *
  /**          ALL RIGHTS RESERVED                                *
  /*******
  /**
  /** Use to stop source spaces if a copy method outside CKZ
  /** is used to copy data sets from the source to the target.
  /**
  /** variables to be filled in ...
  /**
  /** jobcard - job card
  /** srcsys - name of source system
  /** dsnvrl - DB2 high level qualifier
  /** hlq3 - CKZ users high level qualifier
  /** lstdmbr - LISTDEF member name
  /**
  /**
  2 //STOPS EXEC PGM=IKJEFT01
  //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  //SYSPRINT DD SYSOUT=*
  //SYSTSPRT DD SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTPS DD in the source job.
  /**
  3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTPS(lstdmbr)
  4 /** DSN SYSTEM(srss)
  /** -STOP DATABASE(DBASEA) SPACE(TSPACEA)
  /** -STOP DATABASE(DBASEA) SPACE(INDEXA)
  /**
```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-SOURCE-DDN
4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.

Step 5: Set up non-Db2 Cloning Tool Table Space Cloning job to execute data set copies outside Db2 Cloning Tool Table Space Cloning

Employ whatever procedures and syntax appropriate to your chosen replication tool.

If requested by the COPY parameter DATASETS-TO-COPY-DDN, Db2 Cloning Tool Table Space Cloning writes out a list of the source and target VSAM object pairs. The target data set names will have the correct high level qualifier and the correct switch (I/J) indicator if target table spaces and index spaces exist or F0001 if target table spaces and index spaces do not exist. In addition, only the qualifying data set pairs will be listed.

For table spaces and index spaces that do not exist, we recommend that the name(s) of the target Db2 VSAM data set(s) be something that Db2 will not recognize.

Note: After creating the Db2 table spaces and index spaces (either manually or using a tool), we recommend that you change the first character of the fifth node of the data set name (I0001 or J0001) from an “I” or “J” to an “F”. Note that in the Db2 Cloning Tool Table Space Cloning data set copy list, this name is created automatically for the missing targets. If this naming convention is used, then the supplied CKZFIX job will make the data sets available to the target Db2. CKZFIX is documented in topic [“Step 4: Set up and execute optional FIX job \(CKZFIX\)”](#) on page 231.

A sample output:

```
* 06330 11:56:36.79 JOBNAME=JMX8SRC JOBID=J0044593
* SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S
*
*
* TABLE SPACE DATA SETS TO BE COPIED
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN2TSSC.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN2TSSC.J0001.A001
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.J0001.A001
```

Step 6 (Optional): Set up job to re-start source Db2 table spaces and index spaces (CKZSTRS)

The purpose of the CKZSTRS job is to facilitate copying data sets outside of Db2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), Db2 Cloning Tool Table Space Cloning writes Db2 start parameters for the source Db2 table spaces and index spaces to a data set pointed to by COPY keyword START-SOURCE-DDN(?).

Sample JCL for the CKZSTRS job is in the product JCL library, member CKZSTRS. Set up the job to point to the same data set as START-SOURCE-DDN.

The CKZSTRS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //jobcard
  /*JOBPARM S=srcsys
  /******
  /*          COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015          *
  /*          ALL RIGHTS RESERVED                                *
  /******
  /*
  /* Use to start source objects if a copy method outside CKZ
  /* is used to copy data sets from the source to the target.
  /*
  /* variables to be filled in ...
  /*
  /* jobcard - job card
  /* srcsys - name of source system
  /* dsnvr1 - DB2 high level qualifier
  /* hlq3 - CKZ users high level qualifier
  /* lstdmbr - LISTDEF member name
  /*
  /*
2 //STARTS EXEC PGM=IKJEFT01
  //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  //SYSPRINT DD SYSOUT=*
  //SYSTSPRT DD SYSOUT=*
  /*
  /*
  /* SYSTSIN DD was CKZSTRS DD in the source job.
  /*
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTRS(lstdmbr)
4 /* DSN SYSTEM(srcss)
  /* -START DATABASE(DBASEA) SPACE(TSPACEA)
  /* -START DATABASE(DBASEA) SPACE(INDEXA)
  /*
```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword START-SOURCE-DDN

4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.

Step 7 (Optional): Create missing target table spaces and index spaces

As the target table spaces and index spaces did not exist when the Db2 Cloning Tool Table Space Cloning source in step 2 was run, there are no STOP commands for the newly created table spaces and index spaces. If needed, at this point another Db2 Cloning Tool Table Space Cloning source job can be run to output these stop commands.

Use PGM(NONE) and no stops will be issued to the existing source spaces. Additionally, delete the IDCAMS-DDN parameter from the source job. If this delete is not performed, the correct IDCAMS-DDN data set will be overwritten. The next step will delete these data sets and rename the copied data sets from F0001 to I0001.

Step 8 (Optional): Set up job to make copied VSAM objects available to target Db2 (CKZFIX)

The purpose of the CKZFIX job is to facilitate making the data sets copied to the target subsystem available to the target Db2.

When Db2 Cloning Tool Table Space Cloning detects missing target Db2 table spaces and index spaces, the output data set names are handled differently and parameters are written to data sets which will stop the missing table spaces and index spaces (after the user creates them), delete the new VSAM objects and rename the VSAM objects copied by Db2 Cloning Tool Table Space Cloning. .

Sample JCL for the FIX job is in the product JCL library, member CKZFIX. Set up the job to point to the same data sets as STOP-TARGET-DDN and IDCAMS-DDN.

The CKZFIX job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //jobcard
  /*JOBPARM S=trgsys
  /*
  /******
  /*      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015          *
  /*      ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use when missing DB2 objects created after the copy.
  /**
  /** variables to be filled in ...
  /**
  /**   jobcard - job card
  /**   trgsys  - name of target system
  /**   dsnvr1 - DB2 high level qualifier
  /**   hlq3   - CKZ users high level qualifier
  /**   lstdmbr - LISTDEF member name
  /**
  /**
  /******
  /** Stop the DB2 created table space(s) AND index(es) *
  /** for the deletion and rename step to follow.      *
  /******
2 //STOP      EXEC PGM=IKJEFT01
  /*
  /**
  /**STEPLIB DD      DISP=SHR,DSN=dsnvr1.SDSNLOAD
  /**SYSPRINT DD     SYSOUT=*
  /**SYSTSPT DD     SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTPT DD in the source job CKZ2SRC
  /**
3 //SYSTSIN DD     DISP=SHR,DSN=hlq3.CMDSSTPT(lstdmbr)
4 /** DSN      SYSTEM(trss)
  /** -STOP   DATABASE(DBASEA) SPACE(TSPACEA)
  /** -STOP   DATABASE(DBASEA) SPACE(INDEXA)
  /**

```

```

//*
//*****
//* Delete the DB2 created VSAM data set(s) and *
//* rename the cloned data set(s) *
//*****
//*
5 //DELRNAME EXEC PGM=IDCAMS,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//*
//*
//* SYSIN DD was CKZIDCAM DD in the source job CKZ2SRC
//*
6 //SYSIN DD DISP=SHR,DSN=h1q3.IDCAMS(1stdmbr)
//*
//* DELETE DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001 -
//* CLUSTER NOERASE PURGE
//* ALTER -
//* DSN081D.DSNDBC.DBASEA.TSPACEA.F0001.A001 -
//* NEWNAME(DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001)
//* ALTER -
//* DSN081D.DSNDBD.DBASEA.TSPACEA.F0001.A001 -
//* NEWNAME(DSN081D.DSNDBD.DBASEA.TSPACEA.I0001.A001)
//*
//*
//*
//*

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN.
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by Db2 Cloning Tool Table Space Cloning in the output data set.
5. Execution of IDCAMS – To delete the VSAM objects created by the target Db2 when the table spaces and index spaces were created and rename the VSAM objects copied from the source Db2.
6. SYSIN – Data set referenced by source job COPY command keyword IDCAMS-DDN.

Step 9: Set up for a re-run of the source job

Re-running the source job at this point will capture the object IDs of the newly created Db2 table spaces and index spaces and update the data sets used as input for the target job..

The COPY command keyword must be DATA-MOVER(PGM(NONE)).

Step 10: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and SYNCDB2 command to make the VSAM objects accessible on the target subsystem. It will optionally stop and start the target table spaces and index spaces.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. A detailed example follows.

Procedure

1. Specify the PARMLIB data set.
Note that the Db2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCP/IP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target and TCP/IP server jobs. See line #3 in the target job example and the associated note 3.
2. Specify the input data set where the Db2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, line 7.

See line #6 in the target job example and the associated note 6.

Target job example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample Db2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
//*JOBPARM S=trgsys
/*****
/**      COPYRIGHT ROCKET SOFTWARE, INC. 2007-2015      *
/**      ALL RIGHTS RESERVED                            *
/*****
/**
/** Scenarios 1-4 - the target job is the same for all 4 scenarios.
/** The only change is the data set specified in CKZSQL and CKZIN.
/** These are output from the source job.
/**
/** The target job must be run on the target DB2 system.
/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** trgsys - name of target system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSDUMP high level qualifier
/** dsnvrl - DB2 high level qualifier
/** parmmb - CKZ PARMLIB member
/** dumpsc - SYSDUMP STORCLAS if needed
/** dumpmc - SYSDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmbr - LISTDEF member name
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmmb)
/**
/**
/** CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD SYSOUT=*
/**
/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG DD SYSOUT=*
/**
/**
/** dump DDs
/**
/**SYSDUMP DD SYSOUT=* Note: SYSDUMP preferred
/**SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
/** STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/** SPACE=(CYL,(50,30),RLSE),
/** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**ABNLIGNR DD DUMMY do not remove if using ABENDAID
/**
/**
/** CKZIN - input commands that will be used by the target
/** job to make the VSAM object(s) accessible on the target
/** DB2 subsystem. Created by the source job.
/**
6 //CKZIN DD DISP=SHR,DSN=hlq3.SYNCDB2(lstdmbr)
/**
/*****
```

```

/** sample control statements as would appear in SYNCDB2 member
/** Note: all lines have /** added in column 1 for JCL compatibility.
/*******
/**
/**CKZIN DD *
/** 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
/** SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
/**
/**SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCAN0(N)
/**
/**
/** TABLE SPACES - COPIED SUCCESSFULLY */
/**
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0002,X'0047, -
/** TSOB,X'0001,X'0046, -
/** TBOB,X'0003,X'0048, -
/** TBOB,X'0008,X'004D, -
/** TBOB,X'000D,X'0052)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0024,X'0013, -
/** TSOB,X'0023,X'0012, -
/** TBOB,X'0025,X'0014, -
/** TBOB,X'002A,X'0019, -
/** TBOB,X'002F,X'001E)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0046,X'0035, -
/** TSOB,X'0045,X'0034, -
/** TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0046,X'0035, -
/** TSOB,X'0045,X'0034, -
/** TBOB,X'0047,X'0036)

```

Note: In this example, the following:

1. Execution of Db2 Cloning Tool Table Space Cloning main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.

4. CKZPRINT - Displays CKZINI tokens, CKZIN control parameters, Db2 SQL execution status, and SYNCDB2 status and START Db2 command status for each data set processed. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
5. CKZLOG - Displays the Db2 commands issued by Db2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each Db2 page access. This DD is not required. If this DD is output to a data set, the data set must be defined with RECFM=VB and LRECL=132.
6. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Chapter 16. Procedures for cloning Db2 table spaces and index spaces

This topic describes different methods for copying table spaces and index spaces from one Db2 subsystem to another.

The procedures described here provide for various situations, such as the data set copy methodology used to copy the table spaces and index spaces, and whether the table spaces and index spaces already exist in the target Db2 catalog.

Choose the appropriate procedure for your situation.

Copy procedure 1: FlashCopy, SnapShot, or TimeFinder/Clone setup - when all target table spaces and index spaces already exist in Db2 catalog

Use the procedure shown in the following table if you want to copy the source table spaces and index spaces to the target Db2 subsystem when the target table spaces and index spaces already exist in the Db2 catalog and you are using FlashCopy, SnapShot, or TimeFinder/Clone.

| |
|--|
| <i>Table 51. Db2 Cloning Tool Table Space Cloning - copy procedure for FlashCopy, SnapShot, or TimeFinder/Clone when objects exist in the Db2 catalog</i> |
| Copy procedure steps |
| “Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog” on page 261. |
| “Step 2 (Optional): Submit TCP/IP server job” on page 261. If the source and target Db2 subsystems are on different LPARs, then submit the TCP/IP server job CKZTCPS. |
| “Step 3 (Optional): Submit source TCP/IP server job” on page 262. If the source and target Db2 subsystems are on different LPARs, then submit the source TCP/IP server job CKZTCPSS. |
| “Step 4: Submit source job” on page 262. |
| “Step 5: Submit target job” on page 262. |

Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog

Ensure that the target Db2 table spaces and index spaces exist in the target Db2 catalog.

Do this manually, use a non-Db2 Cloning Tool Table Space Cloning tool, or execute the Db2 Cloning Tool Table Space Cloning source job. When using the Db2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target Db2 subsystems are on different LPARs.

Db2 Cloning Tool Table Space Cloning uses TCP/IP to read the target catalog and to issue cross-system Db2 STOP and DISPLAY commands. Target Db2 catalog access is necessary to confirm the existence of the target Db2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target Db2 catalog for ID translations.

Step 3 (Optional): Submit source TCP/IP server job

Submit the source TCP/IP server job if it is not currently running.

The source TCP/IP server job is required when you are cloning across LPARs.

Step 4: Submit source job

The next step is to submit the source job.

The source job:

- Connects to the source and target Db2 subsystems
- Selects the table spaces and index spaces to be replicated from the source Db2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Stops the source and target table spaces and index spaces
- Invokes FlashCopy, SnapShot, or TimeFinder/Clone (if available) to replicate the data sets
- Starts the source table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

Step 5: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target Db2. The target job uses the parameters passed to it by the source job to optionally restart the target Db2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to Db2.

This completes the cloning process.

Copy procedure 2: FlashCopy, SnapShot, or TimeFinder/Clone setup - when some or all target table spaces and index spaces do not exist in Db2 catalog

Ideally, all target Db2 table spaces and index spaces will exist before the Db2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy, and there is not enough time to fit creating the target table spaces and index spaces within the window, then the following procedure is available.

You may also use this procedure (shown in the following table) for the table spaces and index spaces that are inadvertently missed. The Db2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. The output to some data sets will vary depending on whether the table spaces and index spaces are missing on the target or not.

| Copy procedure steps |
|--|
| "Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog" on page 261. |
| "Step 2 (Optional): Submit TCP/IP server job" on page 261. If the source and target Db2 subsystems are on different LPARs, then submit the TCP/IP server job CKZTCPS. |
| "Step 3 (Optional): Submit source TCP/IP server job" on page 262. If the source and target Db2 subsystems are on different LPARs, then submit the source TCP/IP server job CKZTCPSS. |
| "Step 4: Submit source job" on page 263. |

Table 52. Db2 Cloning Tool Table Space Cloning - copy procedure for FlashCopy, SnapShot, or TimeFinder/Clone when some or all target objects do not exist in the Db2 catalog (continued)

Copy procedure steps

“Step 5: Create missing target Db2 table spaces and index spaces - manually or using non-Db2 Cloning Tool Table Space Cloning Tool” on page 264.

“Step 6: Submit FIX job on target z/OS system (CKZFIX)” on page 264.

“Step 7: Rerun source job” on page 264. Ensure that the copy command keyword is DATA-MOVER(PGM(NONE)).

“Step 8: Submit target job” on page 264.

Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog

Ensure that the target Db2 table spaces and index spaces exist in the target Db2 catalog.

Do this manually, use a non-Db2 Cloning Tool Table Space Cloning tool, or execute the Db2 Cloning Tool Table Space Cloning source job. When using the Db2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE)) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target Db2 subsystems are on different LPARs.

Db2 Cloning Tool Table Space Cloning uses TCP/IP to read the target catalog and to issue cross-system Db2 STOP and DISPLAY commands. Target Db2 catalog access is necessary to confirm the existence of the target Db2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target Db2 catalog for ID translations.

Step 3 (Optional): Submit source TCP/IP server job

Submit the source TCP/IP server job if it is not currently running.

The source TCP/IP server job is required when you are cloning across LPARs.

Step 4: Submit source job

The next step is to submit the source job.

The source job:

- Connects to the source and target Db2 subsystems
- Selects the table spaces and index spaces to be cloned from the source Db2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Stops the source and existing target table spaces and index spaces
- Invokes FlashCopy, SnapShot, or TimeFinder/Clone (if available) to replicate the data sets
- Starts the source table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

If the source job detects missing Db2 table spaces or index spaces on the target Db2, Db2 Cloning Tool Table Space Cloning will:

- Use as the high level qualifier, of the data set name(s) that it creates, the alias supplied by the DEFVCAT keyword of the COPY command
- Modify the target data set name(s) that Db2 Cloning Tool Table Space Cloning creates. The first character of the fifth node will be changed from “I” or “J” to “F”. Db2 will not recognize the new data

set name; therefore, the data set cannot be mistakenly corrupted when the missing Db2 table spaces or index spaces are created.

Note: Ordinarily, Db2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if Db2 finds a correctly named data set already in existence, Db2 will use that data set and write internal object IDs into it to associate it with the table space or index space that Db2 created. If data already exists in the VSAM data set, then that data may become permanently inaccessible. New data can be inserted into the data set but the old data may not be retrieved.

- Generate IDCAMS parameters to delete the VSAM object(s) which will, later, be created by the target Db2
- Generate IDCAMS parameters to rename the “F” data sets to a name recognizable to Db2.

Step 5: Create missing target Db2 table spaces and index spaces - manually or using non-Db2 Cloning Tool Table Space Cloning Tool

Once the data set copy is done, you can create the missing table spaces, tables, and index spaces for the target Db2. This step provides the missing object IDs Db2 Cloning Tool Table Space Cloning needs to clone the VSAM objects.

The execution of the source job in Step 3 of this procedure cannot write out STOP commands for non-existent target table spaces and index spaces. If STOP commands are required, run another source job at this point with PGM(NONE) to generate STOP commands for those missing target table spaces and index spaces. These stop commands can then be used in [“Step 6: Submit FIX job on target z/OS system \(CKZFIX\)”](#) on page 264.

Step 6: Submit FIX job on target z/OS system (CKZFIX)

The CKZFIX job uses Db2 commands and IDCAMS parameters passed to it by the source job.

The CKZFIX job will:

- Stop the target Db2 table spaces and index spaces (if available)
- Delete the new VSAM object(s) created by Db2
- Rename the copied VSAM objects to the name(s) of the deleted VSAM object(s)

Ordinarily, Db2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if Db2 finds a correctly named data set already in existence, Db2 will use that data set and write internal object IDs into it to associate it with the table space or index space that Db2 created. If data already exists in the VSAM data set, then that data becomes permanently inaccessible. New data can be inserted into the data set but the old data cannot be retrieved.

Making the target data sets unrecognizable to Db2 prevents it from writing the internal identifiers for any newly created Db2 table spaces and index spaces into those data sets. The target Db2 will allocate new data sets and write into them. The new and empty data sets can then be replaced with data sets copied from the source Db2.

Step 7: Rerun source job

During the re-run of the source job, Db2 Cloning Tool Table Space Cloning will update the parameters for the target job to include the new object IDs.

Step 8: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target Db2. The target job uses the parameters passed to it by the source job to optionally restart the target Db2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to Db2.

This completes the cloning process.

Copy procedure 3: All other copy methodologies - All target table spaces and index spaces already exist in Db2 catalog

Use the procedure shown in the table if you want to copy the source table spaces and index spaces to the target Db2 subsystem when the target table spaces and index spaces already exist in the Db2 catalog and you are using a method other than FlashCopy, SnapShot, or TimeFinder/Clone.

| |
|--|
| <i>Table 53. Db2 Cloning Tool Table Space Cloning - copy procedure for other copy methods when all target objects already exist in the Db2 catalog</i> |
| Copy procedure steps |
| “Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog” on page 261. |
| “Step 2 (Optional): Submit TCP/IP server job” on page 261. If the source and target Db2 subsystems are on different LPARs, then submit the TCP/IP server job CKZTCPS. |
| “Step 3 (Optional): Submit source TCP/IP server job” on page 262. If the source and target Db2 subsystems are on different LPARs, then submit the source TCP/IP server job CKZTCPSS. |
| “Step 4: Submit source job” on page 266 to capture the names of the data sets to be copied. It will also create Db2 commands to stop the source and target table spaces and index spaces and to re-start the source table spaces and index spaces. |
| “Step 5: Submit job CKZSTPT to stop target Db2 table spaces and index spaces” on page 266. |
| “Step 6: Submit job CKZSTPS to stop source Db2 table spaces and index spaces” on page 266. |
| “Step 7: Submit non-Db2 Cloning Tool Table Space Cloning job to copy VSAM objects” on page 266. |
| “Step 8: Submit job CKZSTRS to start source Db2 table spaces and index spaces” on page 266. |
| “Step 9: Rerun source job” on page 266. |
| “Step 10: Submit target job” on page 266. |

Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog

Ensure that the target Db2 table spaces and index spaces exist in the target Db2 catalog.

Do this manually, use a non-Db2 Cloning Tool Table Space Cloning tool, or execute the Db2 Cloning Tool Table Space Cloning source job. When using the Db2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target Db2 subsystems are on different LPARs.

Db2 Cloning Tool Table Space Cloning uses TCP/IP to read the target catalog and to issue cross-system Db2 STOP and DISPLAY commands. Target Db2 catalog access is necessary to confirm the existence of the target Db2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target Db2 catalog for ID translations.

Step 3 (Optional): Submit source TCP/IP server job

Submit the source TCP/IP server job if it is not currently running.

The source TCP/IP server job is required when you are cloning across LPARs.

Step 4: Submit source job

The next step is to submit the source job.

When the COPY command keyword is DATA-MOVER(PGM(NONE)) it is assumed that the copy will be done outside of Db2 Cloning Tool Table Space Cloning.

Db2 Cloning Tool Table Space Cloning still determines the source and target Db2 subsystems, selects from the source Db2 catalog the table spaces and index spaces to be cloned, and confirms the existence of the target table spaces and index spaces. It also checks for object incompatibilities between source and target table spaces and index spaces.

To assist the user, Db2 Cloning Tool Table Space Cloning writes the list of source and target data set names provided by the LISTDEF command to a data set. It also writes stop commands for the source and target Db2 table spaces and index spaces and writes start commands for the source table spaces and index spaces to data sets. This information can be used to prepare the non-Db2 Cloning Tool Table Space Cloning copy job.

Step 5: Submit job CKZSTPT to stop target Db2 table spaces and index spaces

Existing target table spaces and index spaces must be stopped so that Db2 will deallocate the VSAM objects. This allows them to be replaced.

Step 6: Submit job CKZSTPS to stop source Db2 table spaces and index spaces

The source table spaces and index spaces must be stopped so that Db2 will deallocate the VSAM objects. This allows them to be copied.

Step 7: Submit non-Db2 Cloning Tool Table Space Cloning job to copy VSAM objects

Any hardware or software replication utility can be used.

Step 8: Submit job CKZSTRS to start source Db2 table spaces and index spaces

Once the data sets are replicated (copied) the source table spaces and index spaces are no longer needed by the cloning process.

Step 9: Rerun source job

If you skipped the running the source job in step 3 of this procedure, then this will actually be the first time that the source job is run.

The source job:

- Connects to the source and target Db2 subsystems
- Selects the table spaces and index spaces to be replicated from the source Db2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

Step 10: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target Db2. The target job uses the parameters passed to it by the source job to optionally restart the target Db2 table spaces and index spaces and

update the internal identifiers in the copied VSAM objects to make them accessible to Db2. This completes the cloning process.

Copy procedure 4: All other copy methodologies - some or all target table spaces and index spaces do not exist in Db2 catalog

Ideally, all target Db2 table spaces and index spaces will exist before the Db2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy, and there is not enough time to fit creating the target table spaces and index spaces within the window, then the following procedure is available. You may also use this procedure for the table spaces and index spaces that are inadvertently missed.

The Db2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. The output to some data sets will vary depending on whether the table spaces and index spaces are missing on the target or not.

The following table lists the copy procedure when some or all target objects do not exist in the Db2 catalog:

| Copy procedure steps |
|--|
| “Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog” on page 261. |
| “Step 2 (Optional): Submit TCP/IP server job” on page 261. If the source and target Db2 subsystems are on different LPARs, then submit the TCP/IP server job CKZTCPS. |
| “Step 3 (Optional): Submit source TCP/IP server job” on page 262. If the source and target Db2 subsystems are on different LPARs, then submit the source TCP/IP server job CKZTCPSS. |
| “Step 3: Submit source job” on page 268. |
| “Step 5: Create missing target Db2 table spaces and index spaces” on page 269. |
| “Step 6: Submit job CKZSTPT to stop target Db2 table spaces and index spaces” on page 269. |
| “Step 7: Submit job CKZSTPS to stop source Db2 table spaces and index spaces” on page 269. |
| “Step 8: Submit non-Db2 Cloning Tool Table Space Cloning job to copy VSAM objects” on page 269. |
| “Step 9: Submit job CKZSTRS to restart source Db2 table spaces and index spaces” on page 269. |
| “Step 10: Submit FIX job on target z/OS system (CKZFIX)” on page 269. |
| “Step 11: Rerun the source job to prepare target job” on page 270. |
| “Step 12: Submit target job” on page 270. |

Step 1: Verify/create target Db2 table spaces and index spaces in target Db2 catalog

Ensure that the target Db2 table spaces and index spaces exist in the target Db2 catalog.

Do this manually, use a non-Db2 Cloning Tool Table Space Cloning tool, or execute the Db2 Cloning Tool Table Space Cloning source job. When using the Db2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). Db2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target Db2 subsystems are on different LPARs.

Db2 Cloning Tool Table Space Cloning uses TCP/IP to read the target catalog and to issue cross-system Db2 STOP and DISPLAY commands. Target Db2 catalog access is necessary to confirm the existence of the target Db2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target Db2 catalog for ID translations.

Step 3 (Optional): Submit source TCP/IP server job

Submit the source TCP/IP server job if it is not currently running.

The source TCP/IP server job is required when you are cloning across LPARs.

Step 3: Submit source job

Db2 Cloning Tool Table Space Cloning can facilitate using methodologies other than FlashCopy, SnapShot, or TimeFinder/Clone to replicate the Db2 data sets. If you are interested in using the information provided by Db2 Cloning Tool Table Space Cloning for this purpose, then submit the source job at this point. The COPY command keyword must be DATA-MOVER(PGM(NONE)). If you are not interested in the information that Db2 Cloning Tool Table Space Cloning can provide, then skip this step. IBM does not recommend skipping this step.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), it is assumed that the copy will be done outside of Db2 Cloning Tool Table Space Cloning. Db2 Cloning Tool Table Space Cloning still determines the source and target Db2 subsystems, selects from the source Db2 catalog the table spaces and index spaces to be cloned, and confirms the existence of the target objects.

To assist the user, Db2 Cloning Tool Table Space Cloning writes the list of source and target data set names provided by the LISTDEF command to a data set. It also writes stop commands for the source and target Db2 table spaces and index spaces, and restart commands for the source table spaces and index spaces, to data sets. This information can be used to prepare the non-Db2 Cloning Tool Table Space Cloning copy job.

If the source job detects missing Db2 table spaces and index spaces on the target Db2, it will:

- Use the alias supplied by the DEFVCAT keyword of the COPY command as the high level qualifier of the data set name(s) that it creates.
- Modify the target data set name(s) that it derives. The first character of the fifth node will be changed from "I" or "J" to "F". If this naming convention is adhered to, then Db2 will not recognize the new data set name; therefore, the data set cannot be mistakenly corrupted when the missing Db2 table spaces and index spaces are created. Furthermore, Db2 Cloning Tool Table Space Cloning will create parameters to assist in renaming the data sets.

Note: Ordinarily, Db2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if it finds a correctly named data set already in existence, it will use that data set and write internal object IDs into it to associate it with the table space or index space that Db2 created. If data already exists in the VSAM data set, then that data becomes permanently inaccessible. New data can be inserted into the data set but the old data cannot be retrieved.

- Generate IDCAMS parameters to delete the VSAM object(s) which will, later, be created by the target Db2,
- Generate IDCAMS parameters to rename the "F" data sets to a name recognizable to Db2.

Step 5: Create missing target Db2 table spaces and index spaces

Once the missing table spaces and index spaces are flagged, you can create the missing table spaces, tables and indexes for the target Db2. This step provides the missing object IDs Db2 Cloning Tool Table Space Cloning needs to make the VSAM objects accessible.

Step 6: Submit job CKZSTPT to stop target Db2 table spaces and index spaces

Existing target table spaces and index spaces must be stopped so that Db2 will deallocate the VSAM objects. This allows them to be replaced.

This step is optional if the copy utility does not require the source VSAM objects to be deallocated and a FUZZY-COPY is acceptable.

Step 7: Submit job CKZSTPS to stop source Db2 table spaces and index spaces

The source table spaces and index spaces must be stopped so that Db2 will deallocate the VSAM objects. This allows them to be copied.

Step 8: Submit non-Db2 Cloning Tool Table Space Cloning job to copy VSAM objects

Any hardware or software replication utility can be used.

Step 9: Submit job CKZSTRS to restart source Db2 table spaces and index spaces

Once the data sets are replicated (copied) the source table spaces and index spaces are no longer needed by the cloning process.

You may start them manually via DDL or use the start commands provided by the tool. This step is not necessary if the source table spaces and index spaces were not stopped.

Step 10: Submit FIX job on target z/OS system (CKZFIX)

The CKZFIX job uses Db2 commands and IDCAMS parameters passed to it by the source job.

The CKZFIX job will:

- Delete the new VSAM object(s) created by Db2
- Rename the copied VSAM objects to the name(s) of the deleted VSAM object(s)

Ordinarily, Db2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if Db2 finds a correctly named data set already in existence, Db2 will use that data set and write internal object IDs into it to associate it with the table space or index space that Db2 created. If data already exists in the VSAM data set, then that data becomes permanently inaccessible. New data can be inserted into the data set but the old data cannot be retrieved.

Making the target data sets unrecognizable to Db2 prevents it from writing the internal identifiers for any newly created Db2 table spaces and index spaces into those data sets. The target Db2 will allocate new data sets and write into them. The new and empty data sets can then be replaced with data sets copied from the source Db2.

Step 11: Rerun the source job to prepare target job

If you skipped the running the source job in step 3 of this procedure, then this will actually be the first time that the source job is run.

The source job:

- Connects to the source and target Db2 subsystems
- Selects the table spaces and index spaces to be replicated from the source Db2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

Step 12: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target Db2. The target job uses the parameters passed to it by the source job to optionally restart the target Db2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to Db2.

This completes the cloning process.

Chapter 17. Using data masking with table space cloning

Data copied from a source object(s) to a target object(s) may be modified during the copy so that the target data in one or more columns may be different from the source data. The changes are made based on masking rules that are enabled during the copy. Examples of fields that you might change are social security numbers, credit card numbers, names and addresses. In addition, Db2 Cloning Tool Table Space Cloning can generate index rebuild jobs for the changed objects as part of the cloning process.

If the copy is not successful or the target object does not exist, then no masking is performed.

You can apply data masking by using the batch interface commands and DDs. Data masking is also available using the ISPF interface.

Types of data masking

Db2 Cloning Tool Table Space Cloning offers the following types of data masking:

- **STATIC:** masks column values with a constant value.
- **MASK:** modifies values by replacing positions within an existing column value with a customized pattern or static value.
- **PATTERN:** modifies values by replacing an existing column value with a pattern; this type allows you to specify the pattern based on the existing value of the column.
- **RANDOM:** masks values using random numeric or date/timestamp data, optionally within a certain range.
- **USEREXIT:** allows the user to specify their own user exit containing data masking rules.
- **SEQUENCE:** masks by generating a sequence of numeric values with a specified increment.
- **SCRAMBLE:** applies a static scrambling algorithm to the column value.
- **CURRENT DATE:** replaces the column values with the current date.
- **CURRENT TIME:** replaces the column values with the current time.
- **CURRENT TIMESTAMP:** replaces the column values with the current timestamp
- **CURRENT USER:** replaces the column values with the current user ID of the owner of the process when it is run.

Objects and data types not supported

Data masking is not supported with compressed table spaces.

The following data types may not be masked using Db2 Cloning Tool Table Space Cloning data masking:

- All graphic types (GRAPHIC, VARGRAPHIC, LONGVARG)
- All LOB types (BLOB, CLOB, DBCLOB)
- XML
- Masks cannot be applied to any column that is defined part of a hash key.

Masking for tables with RI

When using data masking on columns with referential integrity, the only types of supporting masking functions are SCRAMBLE, PATTERN (with `use_sources (Y)`), and USEREXIT. All referentially related columns will have the data mask applied, even if the masking function is only specified for one column. In addition, a user exit applied to a parent or foreign key column must generate repeatable results.

Summary of steps

These steps assume that you have included the indexes in the LISTDEF and you want to allow Db2 Cloning Tool Table Space Cloning to create and submit the jobs to rebuild indexes. For other ways to rebuild indexes, see [Chapter 20, “Options for rebuilding indexes,” on page 297](#).

1. Add keywords to the COPY command.
2. Add a DD to the source job.
3. Create the masking command members.
4. Add and configure the optional parameters to the COPY and SET commands for rebuilding indexes. For more information, see [Chapter 20, “Options for rebuilding indexes,” on page 297](#).
5. Submit the source job.
6. Submit the target job.

Source job changes

Make the following changes to your source job in order to use data masking.

Add keywords to the COPY command

Add the DATA-MASKING, INCLUDE-ALL-RI, and TARGET-JOB-INDEX-REBUILD-DDN keywords to the COPY command.

Create a member containing MASKDEF commands and keywords

The member is to contain MASKDEF commands that identify the table and column to be masked and define the masking rule. The MASKDEF commands must be formatted as follows. Multiple MASKDEF commands can be included in the member.

```
MASKDEF
  TABLENAME(tbname)
  TABLECREATOR(tbcreator)
  COLUMNNAME(colname)
  MASKRULE(maskrule)
```

TABLENAME (tbname)

(Required) Specify the name of the source table that contains the column to be masked.

TABLECREATOR(tbcreator)

Specify the table creator of the source table containing the column to be masked. The DEFAULT-SQLID is used if this keyword is not included.

COLUMNNAME(colname)

(Required) Specify the column name (in the source table) to be masked.

MASKRULE(maskrule)

(Required) Include the mask rule for the column. Refer to [“Specifying the masking rule” on page 273](#) for syntax and detailed information about the mask rule.

Add a new DD for the MASKDEF data set

Add a DD pointing to the MASKDEF member as follows. This DD is required if DATA-MASKING (Y).

```
CKZMSKDF DD DISP=SHR,DSN=h1q.MASKDEF(mbname)
```

Specifying the masking rule

Db2 Cloning Tool Table Space Cloning applies data masks to columns based on the masking rule that you specify in the MASKDEF command.

To prevent errors during data masking, you must know the specific data type of the column that is being masked and ensure that the Db2 Cloning Tool Table Space Cloning can process that data type.

Continuation rules

A rule that requires continuation must only use one continuation character. For example:

```
RULE(STATIC, "long_string_3456789B123456789C123456789D123456789E1-  
23456789F123456789G123456789H123456789I123456789J123456789K12345678")
```

More than one continuation character is not allowed, such as:

```
RULE(STATIC, "long_string_3456789B123456789C12-  
3456789D123456789E1-  
23456789F123456789G123456789H123456789I123456789J123456789K12345678")
```

MASKRULE(STATIC, *numeric* | “*date_or_time*” | “*string*”)

This mask rule defines a constant value to be used.

The value must be placed between quotation marks for DATE, TIME, TIMESTAMP, CHAR, and VARCHAR data types. For NUMERIC data types (SMALLINT, INTEGER, BIGINT, REAL, DOUBLE, DECFLOAT and DECIMAL), the value must be entered without quotation marks. The keyword STATIC can be defined in upper or lower case.

This rule can be used for the following column data types:

- SMALLINT
- INTEGER
- BIGINT
- REAL
- DOUBLE
- DECFLOAT
- DECIMAL
- DATE
- TIME
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE (Db2 10 and later)
- CHAR
- VARCHAR

Note: If the original value in a VARCHAR field is shorter than the provided STATIC value, the STATIC value will be truncated to the length of the original value. For example, if the value in the original VARCHAR field is "TOM", and STATIC value provided for the MASK is "MASKED", the resulting value after data masking processing is "MAS".

Parameters

numeric

This value can be any of the following :

- A binary integer (small integer, integer, or big integer)
- A decimal number

- A floating point number (real, double, or decimal)

date_or_time

This value can be a time, date, timestamp; for Db2 10 and later, it can be a timestamp with time zone.

string

This value can be any alphanumeric string.

Samples

```
MASKRULE(STATIC, 10)
MASKRULE(STATIC, -100)
MASKRULE(STATIC, -7.2e+75)
MASKRULE(STATIC, "2009-02-04")
MASKRULE(STATIC, "2008-12-01-15.30.30")
MASKRULE(STATIC, "2010-10-31-23.59.59.000000000000+10:00")
MASKRULE(STATIC, "123 Division Street")
```

MASKRULE(MASK, “pattern”, start, end)

This mask rule modifies values by replacing positions within an existing value with the specified pattern or static value.

Non-alphanumeric characters can be generated by enclosing them with a backslash (\) when they are specified. For example, \% \ generates %. The keyword MASK can be defined in upper or lower case.

This rule can be used for the following column data types:

- CHAR
- VARCHAR

Parameters

pattern

Use this variable to specify the value to be placed in the position specified in the start and end fields. You can also type a pattern to be evaluated, then placed in the specified position or positions. Patterns can be specified in three ways: character generation, string selector, or static value.

- Character generation: Characters that are enclosed in square brackets force a random selection of one of the enclosed characters. Ranges of characters can be established by using the dash. Ranges can only include alphanumeric characters. Quantification after a symbol or group of symbols determines how many times this expression can be repeated, as shown in the following table:

| <i>Table 55. Declaration examples</i> | |
|---------------------------------------|-------------------------|
| Declaration | Repeat count |
| {n} | Exactly n |
| {m,n} | From m till n inclusive |

In combination with repeated templates, a set of characters can establish a correspondence with real text, such as digit columns, phone numbers, zip codes, HTML page elements, and so on.

A set of possible symbols must be defined in brackets. For example, [abc] allows one of those three characters to appear in the text. [1234567890] allows any of those digits to be used, as shown in the following table:

| <i>Table 56. Character generation examples</i> | |
|--|--|
| Character specification | Evaluates to |
| [Aa] | A or a |
| [abcde] or [a-e] | Any character between a and e, inclusive |

| Character specification | Evaluates to |
|-------------------------|--|
| [0-9] | Any single-digit number |
| [0-9a-z] | Any single-digit number or any lowercase letter |
| [A-Z]{3} | Any three-character uppercase string |
| [24%\-\&] | One of the following: 2, 4, %, -, or &. |
| [AaBb]{5} | Any five-character string containing any combination of the four specified letters. For example: AAbAa or abbBA or AbAbB |
| [Aab2]{1,17} | A string from one to 17 characters in length made up of any combination of the four specified letters. For example: a2baab2A or A or a2baabAbaA22bAb2A |

- String selector: Strings that are enclosed in parentheses and delimited by the pipe (|) character perform a random selection of one of the strings. Character generators can be included in a string selector. Both alphanumeric and non-alphanumeric characters can be generated. Non-alphanumeric characters can be generated by dereferencing them (using a \) when they are specified. The following table shows some string selector examples.

| Character specification | Evaluates to |
|-------------------------|---------------------------|
| (Mrs Mr Ms) | MRs, Mr, or Ms |
| (C[ATO] A[KLR]) | CA, CT, CO, AK, AL, or AR |

- Static Value: Any string of characters that is not enclosed in brackets (for character generation) or parentheses (for string selection), is considered a static value and is concatenated in the order of appearance. A non-alphanumeric character should be preceded by a backslash (\), as shown in the following table:

| Character specification | Evaluates to |
|--------------------------|--------------------------------------|
| (Sir The\ Great Mr) Bill | Sir Bill, The Great Bill, or Mr Bill |

start, end

Type an inclusive range of the positions in the source data that are to be replaced.

Samples

```
MASKRULE(MASK, "[abc]{1}", 1, 2)
MASKRULE(MASK, "[0]{1}", 1, 2)
MASKRULE(MASK, "[0-9]{10}", 1, 10)
MASKRULE(MASK, "[A-Z]{10}", 1, 10)
MASKRULE(MASK, "[a-z]{5}", 1, 5)
MASKRULE(MASK, "[ABCabc123]{1, 10}", 1, 10)
MASKRULE(MASK, "[\-\!\@\#\$\%\&\*\(\)\_\-\|=+]{10}", 1, 10)
MASKRULE(MASK, "[A-Za-z0-9]{1, 10}", 1, 10)
MASKRULE(MASK, "[A-Za-z0-9\-\!\@\#\$\%\&\*\(\)\_\-\|=+]{1, 10}", 1, 10)
MASKRULE(MASK, "(Mrs|Mr|Ms)", 1, 3)
MASKRULE(MASK, "(C[ATO]|A[KLR])", 1, 3)
MASKRULE(MASK, "(Sir|The\ Great|Mr) Bill", 1, 10)
MASKRULE(MASK, "(Sir|The\ Great|Mr) Bill (C[ATO]|A[A-Za-z0-9])", 1, 20)
MASKRULE(MASK, "[Aab2]", {1,17})
MASKRULE(MASK, "(IL|IA|IN)")
MASKRULE(MASK, "(I|L|A|N)")
```

MASKRULE(PATTERN, “pattern”, “use_sources”)

The pattern rule allows you to specify a pattern to be used to generate a value. A variety of patterns can be generated by specifying a formula for the pattern.

The keyword PATTERN can be defined in upper or lower case.

This rule can be used for the following column data types:

- CHAR
- VARCHAR

Parameters

pattern

This parameter defines the pattern that will generate a value. Nested expressions can be used. All other characters are directly inserted into the rule value. There are several different pattern types that can be created. Each type of pattern is created by typing the appropriate pattern in the pattern field.

- Character generation: Characters that are enclosed in square brackets force a random selection of one of the enclosed characters. Ranges of characters can be established by using the dash. Ranges can only include alphanumeric characters. Quantification after a symbol or group of symbols determines how many times this expression can be repeated.

| Declaration | Repeat count |
|-------------|-------------------------|
| {n} | Exactly n |
| {m,n} | From m till n inclusive |

In combination with repeated templates, a set of characters can establish a correspondence with real text, such as digit columns, phone numbers, zip codes, HTML page elements, and so on.

A set of possible symbols must be defined in brackets. For example, [abc] allows one of those three characters to appear in the text. [1234567890] allows any of those digits to be used.

| Character specification | Evaluates to |
|-------------------------|--|
| [Aa] | A or a |
| [abcde] or [a-e] | Any character between a and e, inclusive |
| [0-9] | Any single-digit number |
| [0-9a-z] | Any single-digit number or any lowercase letter |
| [A-Z]{3} | Any three-character uppercase string |
| [24\%\-\&] | One of the following: 2, 4, %, -, or &. |
| [AaBb]{5} | Any five-character string containing any combination of the four specified letters. For example: AAbaA or abbBA or AbAbB |
| [Aab2]{1,17} | A string from one to 17 characters in length made up of any combination of the four specified letters. For example: a2baab2A or A or a2baabAbaA22bAb2A |

- String selector: Strings that are enclosed in parentheses and delimited by the pipe (|) character perform a random selection of one of the strings. Character generators can be included in a string selector. Both alphanumeric and non-alphanumeric characters can be generated. Non-

alphanumeric characters can be generated by dereferencing them (using a \) when they are specified.

| Table 60. String selector examples | |
|------------------------------------|---------------------------|
| Character specification | Evaluates to |
| (Mrs Mr Ms) | Mrs, Mr, or Ms |
| (C[ATO] A[KLR]) | CA, CT, CO, AK, AL, or AR |

- **Static Value:** Any string of characters that is not enclosed in brackets (for character generation) or parentheses (for string selection), is considered a static value and is concatenated in the order of appearance. A non-alphanumeric character should be preceded by a backslash (\).

| Table 61. Static value examples | |
|---------------------------------|--------------------------------------|
| Character specification | Evaluates to |
| (Sir The\ Great Mr) Bill | Sir Bill, The Great Bill, or Mr Bill |

use_sources

Valid values for this option are Y(ES) or N(O). If this option is Y(es), output data will be generated based on the current value of the field. This function is based on static formula; for any given input value and identical pattern, the output value will be the same. For example, for the pattern:

```
[0-2][0-9]{3}\-[0-9]{4}\-[A-D]{4}
```

and the field is 5985-9597-BDHF, the output result will be always be 2896-1648-ABCD.

Samples

```
MASKRULE(PATTERN, "[0-2][0-9]{3}\-[0-9]{4}\-[A-D]{4}", "Y")
MASKRULE(PATTERN, "[0-2][0-9]{3}\-[0-9]{4}\-[A-D]{4}", Y)
MASKRULE(PATTERN, "[abc]{1}", YES)
MASKRULE(PATTERN, "[0]{1}", Y)
MASKRULE(PATTERN, "[0-9]{10}", NO)
MASKRULE(PATTERN, "[A-Z]{10}", N)
MASKRULE(PATTERN, "[a-z]{5}", YES)
MASKRULE(PATTERN, "[ABCabc123]{1, 10}", Y)
MASKRULE(PATTERN, "[\~\`!\@\#\$\%\&*\(\)\_\-\=\+]{10}", NO)
MASKRULE(PATTERN, "[A-Za-z0-9]{1, 10}", N)
MASKRULE(PATTERN, "[A-Za-z0-9\~\`!\@\#\$\%\&*\(\)\_\-\=\+]{1, 10}", YES)
MASKRULE(PATTERN, "(Mrs|Mr|Ms)", Y)
MASKRULE(PATTERN, "(C[ATO]|A[KLR])", NO)
MASKRULE(PATTERN, "(Sir|The\ Great|Mr) Bill", N)
MASKRULE(PATTERN, "(Sir|The\ Great|Mr) Bill (C[ATO]|A[A-Za-z0-9])", YES)
```

MASKRULE(RANDOM, "min", "max", "check_bounds") MASKRULE(RANDOM, "min_timestamp", "max_timestamp", "min_timezone", "max_timezone", "check_bounds")

The random rule produces random numeric or date/timestamp data within certain bounds.

The keyword RANDOM can be defined in upper or lower case.

This rule can be used for the following column data types:

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- DATE
- TIME

- TIMESTAMP
- TIMESTAMP WITH TIMEZONE (Db2 10 and later)

Parameters

min, max

min and max must be entered between quotation marks for DATE, TIME and TIMESTAMP data types. All numeric values must be entered without quotation marks. These values can be any of the following, but both min and max must be defined and must be the same data type:

- A binary integer (small integer, integer, or big integer)
- A decimal number
- A floating point number (real or double)
- A time, date, or a timestamp

Minimum and maximum values must be specified in the same data type format of the table column data type. For example, if the column type where the mask is to be applied is decimal, then minimum and maximum should be defined in format NNNNN.MM, where precision and scale also correspond to the column data type

If min and max do not correspond to the column data type, Db2 Cloning Tool Table Space Cloning returns an error.

For *min*, enter the lowest possible value to be generated by this rule. For *max*, enter the highest possible value to be generated by this rule.

min_timestampz, max_timestampz, min_timezone, max_timezone

min_timestampz and max_timestampz are used to calculate minimum and maximum time in UTC. min_timestampz after conversion to UTC must be less than max_timestampz after conversion to UTC.

min_timezone and max_timezone will be used to calculate a random TIMEZONE for the final random value TIMESTAMP WITH TIMEZONE. These parameters are optional; if not included, the system time zone of the target system will be used.

After calculation, random value boundaries (min_timestampz (UTC) + min_timezone) and (max_timestampz (UTC) + max_timezone) must be in the range "0001-01-01-00.00.00.000000000000+00:00" to "9999-12-31-23.59.59.999999999999+00:00" inclusive. Otherwise, a range error message will be displayed.

check_bounds

Valid values for this parameter are Y(ES) or N(O). Enter Y or YES to have one row in your target table that corresponds to the value specified in the min field and one row in your table that corresponds to the value specified in the max field. This option allows you to test the endpoints of the range that you specified.

Samples

```
MASKRULE(RANDOM, 0, 1, N)
MASKRULE(RANDOM, -1, 0, NO)
MASKRULE(RANDOM, -100, 0, Y)
MASKRULE(RANDOM, -1, 1, YES)
MASKRULE(RANDOM, 0.00, 100.00, NO)
MASKRULE(RANDOM, -100.00, 0.00, Y)
MASKRULE(RANDOM, "2008-01-01", "2008-12-31", Y)
MASKRULE(RANDOM, "2008-12-01", "2009-01-31", N)
MASKRULE(RANDOM, "00.00.00", "23.59.59", Y)
MASKRULE(RANDOM, "12.00.00", "12.59.59", N)
MASKRULE(RANDOM, "2008-01-01-00.00.00", "2008-12-31-23.59.59.999999", Y)
MASKRULE(RANDOM, "2008-12-01-12.00.00", "2009-01-31-12.59.59.999999", N)
MASKRULE(RANDOM, "2008-01-01-00.00.00+10:00",
"2008-12-31-23.59.59.9999999999-10:00", "-05:00", "+12:30", Y)
```

MASKRULE(USEREXIT, *module*)

The user exit rule allows Db2 Cloning Tool Table Space Cloning to call an exit that defines the user's own data masking rules.

The keyword USEREXIT can be defined in upper or lower case.

This rule can be used for any column data type.

Parameters

module

For *module*, enter the external module name that will be called for each column value. The current value of the column will be used as input for the module. The user exit should change the value and return it to Db2 Cloning Tool Table Space Cloning in the same buffer. The user exit must reside in the STEPLIB concatenation.

The length of the column values cannot be changed using a user exit.

The following format is used to pass the column value to the user exit:

```
int user_exit(void* field_data, size_t length);
```

Samples

```
MASKRULE(USEREXIT, CSNMASK1)
```

MASKRULE(SEQUENCE, *initial_value*, *increment*)

The sequence rule generates a sequence of numeric values.

The first time the function reference is evaluated, it returns the value of the first argument (*initial_value*). Each subsequent evaluation returns the value of the second argument (*increment*) added to the previously returned value. The resulting data type is always integer.

The keyword SEQUENCE can be defined in upper or lower case.

This rule can be used for the following column data types:

- SMALLINT
- INTEGER
- BIGINT

Parameters

initial_value

Enter an integer value that will be the first value in the sequence.

increment

Enter an integer value that will be increment each value in the sequence.

Samples

```
MASKRULE(SEQUENCE, 2789, 2)
MASKRULE(SEQUENCE, 100, 10)
MASKRULE(SEQUENCE, -100, 10)
```

MASKRULE(SCRAMBLE)

The scramble rule applies a predefined Db2 Cloning Tool Table Space Cloning scrambling function to the value specified by the argument. The result has the same data type as the provided argument.

The keyword SCRAMBLE can be defined in upper or lower case.

This rule can be used for the following column data types:

- SMALLINT
- INTEGER
- BIGINT
- REAL
- DOUBLE
- DECFLOAT
- DECIMAL
- DATE
- TIME
- TIMESTAMP
- CHAR
- VARCHAR
- BINARY
- VARBINARY

Parameters

None.

Samples

```
MASKRULE (SCRAMBLE)
```

MASKRULE(CURRENT DATE)

MASKRULE(CURRENT TIME)

MASKRULE(CURRENT TIMESTAMP, "*timezone*")

These date and time-related rules replace the column values with the current date, current time or current timestamp values.

The keywords CURRENT DATE, CURRENT TIME, and CURRENT TIMESTAMP can be defined in upper or lower case.

This rule can be used for the following column data types:

- DATE
- TIME
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE (Db2 10 and later)

Parameters

timezone

For Db2 10 and later, timezone can be specified with CURRENT TIMESTAMP. Valid values are from -12:59 to +14:00. If the value is omitted, then the system time zone of the target system will be used.

Samples

```
MASKRULE (CURRENT DATE)
MASKRULE (CURRENT TIME)
MASKRULE (CURRENT TIMESTAMP)
MASKRULE (CURRENT TIMESTAMP, "-04:00")
```

MASKRULE(CURRENT USER)

The current user rule replaces the column value with the current user ID of the owner of the process when it is run.

The keyword CURRENT USER can be defined in upper or lower case.

This rule can be used for the following column data types:

- CHAR
- VARCHAR

Parameters

None.

Samples

```
MASKRULE(CURRENT USER)
```

Target job changes

After the source job is executed, the target job will contain the MASKCMD command added to the SYNCDB2 data set. MASKCMD includes all the information from the source job MASKDEF command (read from the source job maskdef DD), as well as the data set name and whether all partitions are included.

MASKCMD example

After the source job is executed, the target job will contain the MASKCMD command. An example follows:

```
MASKCMD TABLENAME(tbname)
        TABLECREATOR(tbcreator)
        COLNAME(colname)
        MASKRULE(maskrule)
        DSN(dsn)
        ALLPARTS(Y|N)
```

In addition, there is one TABLEDEF command for each table with a masked column in a successfully copied data set (or data sets, if partitioned). For tables with RI, there must be at least one PARENT and CHILD subcommand of TABLEDEF for each column with an RI relationship. For example:

```
TABLEDEF
(TBCREATOR(creator)           -
 TBNAME(name)                 -
 DBNAME(name)                 -      DB table resides in
 TSNAME(name)                 -      TS table resides in
 CCSID(char)                  -      encoding scheme of the table
 OBID(x'n)                    -      target table object ID
 REL(Y|N)                     -      if Y, rels exist for this table
 COLNO(X'number of columns)   -      # of columns in this table
 PARENTNO(X'number of parents)-      # of parents
 CHILDNO(X'number of children)-      # of children
 COLUMN(name,                 column name
 coltype,                    column type (CHAR, INT, etc.)
 colno,                      number of column in this table
 length,                    column length
 scale,                      scale if decimal column
 nulls,                      Y or N (nulls exist)
 ccsid                       column CCSID in hex

{,PARENT,
 creator -
 ,table -
 ,relname -
 ,# of columns -
 ,column -
 . . .
 ,column_n -
 . . .
```



```

MASKCMD -
  TBCR (PDDONAA) -
  TBNM (Z@63TB11) -
  CLNM (COL1Z@63TB11CHAR) -
  RULE (STATIC,"3458-08-06") -
  DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
  ALLP (Y)

```

```

TABLEDEF -
  TBCREATOR (PDDONAA) -
  TBNAME (Z@63TB11) -
  DBNAME (Z@63DB) -
  TSNAME (Z@63TS1) -
  CCSID (E) -
  OBID (X'0003) -
  REL (N) -
  COLNO (X'0006) -
  PARENTNO (X'0000) -
  CHILDNO (X'0001) -
  COLUMN (
    COL1Z@63TB11CHAR -
    ,CHAR -
    ,X'0001 -
    ,X'0028 -
    ,X'0000 -
    ,N -
    ,X'00000025 -
    COL2Z@63TB11CHAR2 -
    ,CHAR -
    ,X'0002 -
    ,X'000A -
    ,X'0000 -
    ,N -
    ,X'00000025 -
    COL3Z@63TB11IDENT -
    ,INTEGER -
    ,X'0003 -
    ,X'0004 -
    ,X'0000 -
    ,N -
    ,X'00000000 -
    COL4Z@63TB11ROWID -
    ,ROWID -
    ,X'0004 -
    ,X'0011 -
    ,X'0000 -
    ,N -
    ,X'00000000 -
    COL5Z@63TB11SMALLI -
    ,SMALLINT -
    ,X'0005 -
    ,X'0002 -
    ,X'0000 -
    ,X'00000025 -
    COL3Z@63TB11IDENT -
    ,INTEGER -
    ,X'0003 -
    ,X'0004 -
    ,X'0000 -
    ,N -
    ,X'00000000 -
    COL4Z@63TB11ROWID -
    ,ROWID -
    ,X'0004 -
    ,X'0011 -
    ,X'0000 -
    ,N -
    ,X'00000000 -
    COL5Z@63TB11SMALLI -
    ,SMALLINT -
    ,X'0005 -
    ,X'0002 -
    ,X'0000 -
    ,Y -
    ,X'00000000 -
    COL6Z@63TB11INT -
    ,INTEGER -
    ,X'0006 -
    ,X'0004 -
    ,X'0000 -
    ,Y -
  )

```

```

,X'00000000 -
,CHILD, -
PDDONAA -
,Z@63TB21 -
,COL3Z@63 -
,X'0001 -
,COL3Z@63TB21IDENT -
)

MASKCMD -
TBCR (PDDONAA) -
TBNM (Z@63TB11) -
CLNM (COL2Z@63TB11CHAR2) -
RULE (MASK,"[A-Za-z0-9]{1, 10}") -
DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
ALLP (Y)

MASKCMD -
TBCR (PDDONAA) -
TBNM (Z@63TB11) -
CLNM (COL6Z@63TB11INT) -
RULE (RANDOM, 1, 100, YES) -
DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
ALLP (Y)

MASKCMD -
TBCR (PDDONAA) -
TBNM (Z@63TB11) -
CLNM (COL5Z@63TB11SMALLI) -
RULE (SEQUENCE, 1, 1) -
DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
ALLP (Y)

TABLEDEF -
TBCREATOR (PDDONAA) -
TBNAME (Z@63TB21) -
DBNAME (Z@63DB) -
TSNAME (Z@63TS2) -
CCSID (E) -
OBID (X'0016) -
REL (Y) -
COLNO (X'0005) -
PARENTNO (X'0001) -
CHILDNO (X'0004) -
COLUMN (
COL1Z@63TB21CHAR -
,CHAR -
,X'0001 -
,X'0028 -
,X'0000 -
,N -
,X'00000025 -
,COL2Z@63TB21CLOB -
,CLOB -
,X'0002 -
,X'0004 -
,X'0000 -
,Y -
,X'00000025 -
,COL3Z@63TB21IDENT -
,INTEGER -
,X'0003 -
,X'0004 -
,X'0000 -
,N -
,X'00000000 -
,COL4Z@63TB21ROWID -
,ROWID -
,X'0004 -
,X'0011 -
,X'0000 -
,N -
,X'00000000 -
,COL5Z@63TB21SMALLI -
,SMALLINT -
,X'0005 -
,X'0002 -
,X'0000 -
,N -
,X'00000000 -
,PARENT, -
PDDONAA -

```

```

,Z@63TB11 -
,COL3Z@63 -
,X'0001 -
,COL3Z@63TB11IDENT -
,CHILD, -
,PDDONAA -
,Z@63TB41 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB41SMALLI -
,PDDONAA -
,Z@63TB41 -
,COL1Z@63 -
,X'0002 -
,COL1Z@63TB41CHAR -
,COL5Z@63TB41SMALLI -
,PDDONAA -
,Z@63TB51 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB51SMALLI -
,PDDONAA -
,Z@63TB52 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB52SMALLI -
)

```

```

TABLEDEF -
TBCREATOR (PDDONAA) -
TBNAME (Z@63TB41) -
DBNAME (Z@63DB) -
TSNAME (Z@63TS4) -
CCSID (E) -
QBID (X'004B) -
REL (Y) -
COLNO (X'0005) -
PARENTNO (X'0002) -
CHILDNO (X'0000) -
COLUMN (
COL1Z@63TB41CHAR -
,CHAR -
,X'0001 -
,X'0028 -
,X'0000 -
,Y -
,X'00000025 -
,COL2Z@63TB41CLOB -
,CLOB -
,X'0002 -
,X'0004 -
,X'0000 -
,Y -
,X'00000025 -
,COL3Z@63TB41IDENT -
,INTEGER -
,X'0003 -
,X'0004 -
,X'0000 -
,N -
,X'00000000 -
,COL4Z@63TB41ROWID -
,ROWID -
,X'0004 -
,X'0011 -
,X'0000 -
,N -
,X'00000000 -
,COL5Z@63TB41SMALLI -
,SMALLINT -
,X'0005 -
,X'0002 -
,X'0000 -
,N -
,X'00000000 -
,PARENT, -
,PDDONAA -
,Z@63TB21 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB21SMALLI -
,PDDONAA -

```

```

,Z@63TB21 -
,COL1Z@63 -
,X'0002 -
,COL1Z@63TB21CHAR -
,COL5Z@63TB21SMALLI -
)
TABLEDEF -
TBCREATOR (PDDONAA) -
TBNAME (Z@63TB51) -
DBNAME (Z@63DB) -
TSNAME (Z@63TS5) -
CCSID (E) -
OBID (X'005C) -
REL (Y) -
COLNO (X'0005) -
PARENTNO (X'0001) -
CHILDNO (X'0000) -
COLUMN (
, COL1Z@63TB51CHAR -
, CHAR -
, X'0001 -
, X'0028 -
, X'0000 -
, Y -
, X'00000025 -
, COL2Z@63TB51CLOB -
, CLOB -
COLNO (X'0005) -
PARENTNO (X'0001) -
CHILDNO (X'0000) -
COLUMN (
, COL1Z@63TB51CHAR -
, CHAR -
, X'0001 -
, X'0028 -
, X'0000 -
, Y -
, X'00000025 -
, COL2Z@63TB51CLOB -
, CLOB -
, X'0002 -
, X'0004 -
, X'0000 -
, Y -
, X'00000025 -
, COL3Z@63TB51IDENT -
, INTEGER -
, X'0003 -
, X'0004 -
, X'0000 -
, N -
, X'00000000 -
, COL4Z@63TB51ROWID -
, ROWID -
, X'0004 -
, X'0011 -
, X'0000 -
, N -
, X'00000000 -
, COL5Z@63TB51SMALLI -
, SMALLINT -
, X'0005 -
, X'0002 -
, X'0000 -
, N -
, X'00000000 -
,PARENT, -
PDDONAA -
,Z@63TB21 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB21SMALLI -
)
TABLEDEF -
TBCREATOR (PDDONAA) -
TBNAME (Z@63TB52) -
DBNAME (Z@63DB) -
TSNAME (Z@63TS5) -
CCSID (E) -
OBID (X'0066) -

```

```

REL          (Y) -
COLNO       (X'0005) -
PARENTNO    (X'0001) -
CHILDNO     (X'0000) -
COLUMN      (
      COL1Z@63TB52CHAR -
      ,CHAR -
      ,X'0001 -
      ,X'0028 -
      ,X'0000 -
      ,Y -
      ,X'00000025 -
      ,COL2Z@63TB52CHAR16 -
      ,CHAR -
      ,X'0002 -
      ,X'0010 -
      ,X'0000 -
      ,Y -
      ,X'00000025 -
      ,COL3Z@63TB52IDENT -
      ,INTEGER -
      ,X'0003 -
      ,X'0004 -
      ,X'0000 -
      ,N -
      ,X'00000000 -
      ,COL4Z@63TB52ROWID -
      ,ROWID -
      ,X'0004 -
      ,X'0011 -
      ,X'0000 -
      ,N -
      ,X'00000000 -
      ,COL5Z@63TB52SMALLI -
      ,SMALLINT -
      ,X'0005 -
      ,X'0002 -
      ,X'0000 -
      ,N -
      ,X'00000000 -
      ,PARENT, -
      PDDONAA -
      ,Z@63TB21 -
      ,COL5Z@63 -
      ,X'0001 -
      ,COL5Z@63TB21SMALLI -
      )

```

```

/* INDEXSPACES - COPIED SUCCESSFULLY */

```

```

SYNCDB2 INDEXSPACE COMMANDS

```

```

-
```

Chapter 18. Using image copies to clone table spaces and index spaces

You can use Db2 image copies as the source for cloning table spaces and indexes. Db2 Cloning Tool Table Space Cloning reads the appropriate image copies, reads and applies any necessary log records, writes the target data sets, and updates the data pages to make them usable by the target Db2. This feature eliminates the need to access source objects and allows you to clone objects or an entire application at a specific point in time.

Advantages of cloning from image copies include the following:

- You can select various consistent points to clone to (for example, TO_CURRENT, TO_QUIESCE, TO_LOGPOINT, or TO_TIMESTAMP), making it easy to clone a set of Db2 objects to a particular point in time.
- When you use image copies as the source of the cloning, the source table spaces and index spaces are not accessed.
- If your site does not have a fast replication tool or you cannot use your fast replication tool for cloning, using image copies as the source can provide a faster cloning process than using traditional “slow” copy procedures.

Db2 Cloning Tool Table Space Cloning's source job for this function differs from the source jobs that do not use image copies as the source. When cloning from image copies, the source job does most of the processing. This includes:

- Locating and reading the last full image copy (both SHRLEVEL REFERENCE and SHRLEVEL CHANGE image copies are supported)
- Applying incremental image copies, if they exist
- Reading and applying log records to the image copy clones to bring the clones to a specific consistent point in time
- Performing updates to make the data usable by the target objects
- Writing the data sets for the target objects

The target job can optionally rebuild indexes and perform other processing that is related to specified cloning options.

Objects that are supported for cloning from image copies are table spaces, index spaces, and LOBs. Indexes can be cloned if image copies are available, or they can be rebuilt.

Restrictions

- Source and target objects must match, or target objects must not exist and must be created by Db2 Cloning Tool Table Space Cloning. Db2 Cloning Tool Table Space Cloning does not verify that the image copy object structure matches the source or target catalog, but checks for mismatches by comparing the source and target catalog information. If the objects in the image copy structure do not match the target catalog, an abend may occur after the cloning process (data manager error or other Db2 abend).
- If a table space was altered to NOT LOGGED between the image copy and the user-specified endpoint, logs can only be applied to the point at which the ALTER was issued.
- Data masking is not supported.
- Cloning directly from a system-level backup (SLB) is not supported. However, you can create image copies from SLBs by using Db2 Recovery Expert, then use those image copies as the source for the clone.
- XML processing is not supported.
- DDL can be generated during cloning from image copies, but the DDL is created from the current catalog definitions.

- The process to clone from image copies cannot use subtasks to multi-task multi-data set non-partitioned objects; therefore, SUBTASKS-DATASET-EXTENSIONS is not allowed.
- The following additional commands are not supported: SIM(A), SUBTASK-DATASET-EXTENSIONS, and any of the DATA-MOVER subcommands such as FASTREP or FCTOPPRCPRIARY.
- Cloning of PBR RPN table spaces from image copies is not supported.

System requirements

This functionality requires LOG-APPLY processing. During customization of Db2 Cloning Tool Table Space Cloning, you must select the required tasks and steps to create the objects that are needed for LOG-APPLY.

Steps for cloning from image copies (including index rebuilds)

Follow these steps to clone from image copies and rebuild indexes as part of the cloning process.

Before you begin

If you use the ISPF interface to build and generate the cloning jobs, you do not need to follow the steps in this procedure; Db2 Cloning Tool includes the required syntax and DDs as part of the generation process. However, you should ensure that the source subsystem Db2 SDSNEXIT library is in the STEPLIB for the source job, and you should review the following steps for other requirements and index rebuild instructions.

About this task

These steps assume that:

- You have already set up the basic source and target jobs that contain the control statements to identify the source and target objects and the PARMLIB data sets. Refer to Chapter 15, “Set up procedures for copy by data set for all other methodologies,” on page 237 for information about setting up the jobs.
- You have included the indexes in the LISTDEF and you want to allow Db2 Cloning Tool Table Space Cloning to create and submit index rebuilds. For other ways to rebuild indexes, see Chapter 20, “Options for rebuilding indexes,” on page 297.

Procedure

1. Make the following source job changes:
 - Add DATAMOVER(PGM(SRCIMCPY) and its required parameters to the COPY command.
 - Add LOG-APPLY and the required parameters for cloning from source image copies to the COPY command.
 - Add DDs to the source job as specified in “Source job changes for cloning from image copies” on page 292.
 - Ensure that the source subsystem Db2 SDSNEXIT library is in the STEPLIB for the source job.
 - Add or set up the optional parameters to the COPY and/or SET commands for rebuilding indexes. For more information, see Chapter 20, “Options for rebuilding indexes,” on page 297.
2. Submit the following jobs, depending on whether you are cloning within the same or different LPARs:
 - If you are cloning within the same LPAR:
 - a. Submit the source job.
 - b. Submit the target job.
 - If you are cloning across multiple LPARs:
 - a. Submit the target TCP/IP server job (if not currently running). The target TCP/IP server job is required when cloning across LPARs.

- b. Submit the source TCP/IP server job (if not currently running). The source TCP/IP server job is required when cloning across LPARs and LOG-APPLY is enabled.
- c. Submit the source job.
- d. Submit the target job.

Selecting a specific image copy to use as the cloning source

When you specify the end point for the clone, Db2 Cloning Tool Table Space Cloning selects the most recent image copy before that end point as the source for the cloning. However, you might want to select a specific image copy to clone from (and possibly apply logs after that point). Follows these steps to select a specific full image copy as the source for the cloning.

Procedure

1. Query the SYSIBM.SYSCOPY table to find the START_RBA (RBA or LRSN) value (in hexadecimal) of the desired full image copy.
2. In the source job, specify END-POINT (TO_LOGPOINT X'*byte_string*'). For the *byte_string* value, add 1 (in hexadecimal) to the start RBA or LRSN value of the desired image copy.

Note: If you want Db2 Cloning Tool Table Space Cloning to apply log records that were written after the selected image copy, specify a start RBA between the desired image copy and the next image copy.

Results

When the source job is run, Db2 Cloning Tool Table Space Cloning starts with the end point that was provided, scans backwards to locate the image copy, and applies any logs if the end point requires it.

Example

The following sample query could be used to produce a list of image copies and quiesce points for a table space:

```
SELECT
    HEX(START_RBA) AS START_RBA
    , SUBSTR(DSNAME, 1, 44) AS DSNAME
    , ICTYPE
    , DBNAME
    , TSNAME
    , TIMESTAMP
    , SHRLEVEL
    , OTYPE
FROM SYSIBM.SYSCOPY
WHERE ( DBNAME LIKE 'dbname1%'
        AND TSNAME IN ('tsname1'
                       , 'indexname1'
                       , 'tsname2' )
        AND ( SHRLEVEL IN ('R','C') OR ICTYPE = 'Q' ) )
ORDER BY OTYPE DESC, TIMESTAMP
```

In the output, the START_RBA column contains the start RBA value in hexadecimal. To select the last full image copy, find the desired image copy in the output and note the start RBA value. Then, add 1 (in hexadecimal) to the start RBA value and include it in the END-POINT parameter.

For example, if the START_RBA of the desired image copy is 00C2E5794C49, specify END-POINT (TO_LOGPOINT X'00C2E5794C4A').

To apply logs, specify an end point between 00C2E5794C49 and the start RBA of the next image copy.

Selecting the last image copy to use as the cloning source

If you simply want to clone from the last image copy, you can use a combination of LOG-APPLY parameters to select the last image copy and bypass applying the logs.

Procedure

- In the source job, specify the LOG-APPLY parameters END-POINT(TO_CURRENT) and SKIP-LOG-APPLY(Y).



Attention: To ensure consistency in the target Db2 objects, the last image copies must be consistent FlashCopy image copies that were created for all cloned objects in one call of the COPY utility.

Results

When the source job is run, Db2 Cloning Tool Table Space Cloning starts with the current time and scans backwards to locate the last image copy. That image copy is used as the source for the clone.

Source job changes for cloning from image copies

Make the following changes to your source job in order to clone from image copies.

- Add the DATAMOVER(PMG(SRCIMCPY) keyword to the COPY command.
- Add the LOG-APPLY subcommand with the LA-ENABLE(Y) and ZPARAM-MEMBER keywords to the COPY command. To specify a consistent point and other parameters, refer to [“COPY command syntax” on page 514](#) for details.
- Add the following DDs to the source job:

```
- //BSDS01 DD DISP=SHR,DSN=h1q.BSDS01
  //BSDS02 DD DISP=SHR,DSN=h1q.BSDS02
```

where *h1q* should be set to the high level qualifier for the source subsystem.

- Add a SYSOUT DD to contain output messages, such as //SYSOUT DD SYSOUT=*
- Add the following SYSINCKZ DD as shown (required for LOG-APPLY):

```
SYSINCKZ DD DISP=SHR,DSN=h1q.LOGAPCTL(membername)
```

- Ensure that the Db2 SDSNEXIT library is in the STEPLIB for the source job.

Chapter 19. Using LOG-APPLY to make consistent copies of table spaces and index spaces

The LOG-APPLY keyword allows the Db2 log records written to the source objects to be applied to the target objects during target job processing.

Any updates that are made to the source objects in the time between submitting the source job and running the target job are applied to the target objects. LOG-APPLY provides you with a way to consistently clone a set of related Db2 objects from source table spaces to target table spaces without stopping them or making them unavailable for updates. The LOG-APPLY procedure varies depending on whether you are cloning between or within subsystems on the same LPAR of a z/OS sysplex, or if you are cloning between Db2 subsystems on different LPARs or on different z/OS sysplexes.

- When cloning across multiple LPARs, LOG-APPLY runs on the target job after the source job has completed. A target TCP/IP server job is required for the source job, and a source TCP/IP server job is required for the target job. The target TCP/IP server job connects to the target Db2 subsystem; the source TCP/IP server job connects to the source Db2 subsystem.
- When cloning within the same or different LPAR, you can use LOG-APPLY to copy image copy data sets from the source subsystem to the target subsystem. The target TCP/IP server must be specified if copying across LPARs.
- When cloning within the same LPAR, the LOG-APPLY process runs in the target job, but does not need a source TCP/IP server job.

You can minimize the amount of time it takes to process Db2 logs by specifying a QUIESCE as part of LOG-APPLY and by running the target job in a timely manner.

Restrictions

- Log apply processes that require additional data sets or additional non-SMS managed volumes are not supported.
- Db2 Cloning Tool manages extent size during log apply processing, and does not verify the value of the subsystem parameter MGEXTSZ, which controls whether secondary extent allocations for Db2-managed data sets are to be sized according to a sliding scale.
- Extent size and allocation for user-defined table and index spaces is controlled by the size that is specified when the data set is allocated. During the log apply process, extents are allocated by VSAM processing.
- Cloning of PBR RPN table spaces using log apply is not supported.

System requirements

During customization of Db2 Cloning Tool, you must select the required tasks and steps to create the objects needed for LOG-APPLY. The table space cloning log apply table MINI_LOG_V14 is required for source and target LOG-APPLY processing, and should be created on the source Db2 subsystem. Refer to [“Worksheets: Gathering parameter values for Tools Customizer” on page 31](#) for more information about creating the table.

Before you begin

If you use the ISPF interface to build and generate the cloning jobs, you do not need to follow the steps in this procedure; Db2 Cloning Tool includes the required syntax and DDs as part of the generation process. However, you should carefully review the following steps for other requirements and instructions for rebuilding indexes.

Summary of steps

These steps assume that you have included the indexes in the LISTDEF and you want to allow Db2 Cloning Tool Table Space Cloning to create and submit the jobs to rebuild indexes. For other ways to rebuild indexes, see [Chapter 20, “Options for rebuilding indexes,” on page 297](#).

1. Add LOG-APPLY and other keywords and their required parameters to the COPY and SET commands.
2. Add DDs to the source job.
3. Add or set up the optional parameters to the COPY and/or SET commands for rebuilding indexes. For more information, see [Chapter 20, “Options for rebuilding indexes,” on page 297](#).
4. Add DDs to the target job.
5. Submit the following jobs, depending on whether you are cloning within the same or different LPARs:
 - If you are cloning within the same LPAR:
 - a. Submit the source job.
 - b. Submit the target job.
 - If you are cloning across multiple LPARs:
 - a. Submit the source and target TCP/IP server jobs (if not currently running). The source TCP/IP server job is required when cloning across LPARs using LOG-APPLY.
 - b. Submit the source job.
 - c. Submit the target job.

JCL library members CKZ5SRC and CKZ5TRG contain examples of setting up a job using log apply.

Source job changes

Make the following changes to your source job in order to apply Db2 logs.

- Add the LOG-APPLY subcommand with the LA-ENABLE(Y) and ZPARAM-MEMBER keywords' and their parameters to the COPY command; refer to [“COPY command syntax” on page 514](#) for information about the keywords and parameters. If you are cloning across multiple LPARs, you must specify the USE-TCPIP (Y) parameter in the LOG-APPLY keyword.
- Add the following DDs to the source job:

```
//BSDS01 DD DISP=SHR,DSN=h1q.BSDS01
//BSDS02 DD DISP=SHR,DSN=h1q.BSDS02
```

where *h1q* should be set to the high level qualifier for the source subsystem.

- Ensure that the source subsystem Db2 SDSNEXIT library is in the STEPLIB for the source job.

Using LOG-APPLY across multiple LPARs

CKZTCPSS, the source TCP/IP server job, is required for LOG-APPLY when the source and target Db2 systems are on different LPARs.

Configure the JCL for the source TCP/IP server job and submit it to run on the same LPAR where the source Db2 subsystem is running. The Db2 Cloning Tool target job connects to the source TCP/IP server job that is running on the same LPAR as the source Db2, and then instructs the server to do the following:

- Connect to the source Db2 subsystem
- Determine the logs records that need to be extracted
- Read those log records, and process them for delivery to the target job
- Send the log records back to the target job via TCP/IP

To use LOG-APPLY for cross-LPAR cloning, ensure the following:

- The source Db2 SDSNEXIT library is in the STEPLIB for the source TCP/IP server job.

- The following DDs are defined in the source TCP/IP server job:

```
//SYSINCKZ DD DISP=SHR,DSN=h1q.LOGAPCTL(membername)
//SYSOUT DD SYSOUT=*
//INFOM DD SYSOUT=*
```

The SYSINCKZ DD for the source TCP/IP server job is not required to point to the same data set as the SYSINCKZ DD of the target job. Also, cross-LPAR cloning scenarios do not require that the target jobs have the source Db2 subsystem SDSNEXIT library in their STEPLIBs.

LOG-APPLY in data sharing environments when one or more members are not available

When data sharing group members are inactive and their ZPARM members and BSDS data sets are not available, follow these instructions to skip the inactive members during log apply processing.

By default, log apply processing must read the ZPARM members and BSDS data sets for all data sharing group members. This may not be acceptable in environments with data sharing members that have not been active for a long time, and when their ZPARM members and BSDS data sets are no longer available. For these environments, Db2 Cloning Tool supports the ability to skip inactive data sharing members. Only inactive data sharing member processing can be skipped.

To use this feature, follow the steps below:

1. For members that no longer have the ZPARM members and BSDS data sets available, remove the member information from the DATA-SHARING-MEMBERS keyword. If using the ISPF interface, this can be done by removing the members from the Db2 control file.
2. Specify ALLOW-SKIP-INACTIVE-MEMBERS(Y) in the LOG-APPLY group of parameters of the application cloning COPY command.

If the ZPARM members and BSDS data sets for inactive members are still available, do not use the ALLOW-SKIP-INACTIVE-MEMBERS(Y) keyword. Instead, list all data sharing members, including inactive members, in the DATA-SHARING-MEMBERS keyword. Listing all members helps ensure that the target objects are available after the cloning.

The table below summarizes Db2 Cloning Tool behavior based on the ALLOW-SKIP-INACTIVE-MEMBER and DATA-SHARING-MEMBERS keywords and the status of an excluded member.

Table 62. Copy outcomes depending on data sharing group member status and keyword values

| ALLOW-SKIP-INACTIVE-MEMBER value | Member included in DATA-SHARING-MEMBERS? | Member status | Result |
|---|---|----------------------|---|
| N | Y | Active | Copy is allowed. |
| | | Other than active | Copy is allowed. |
| | N | Active | Copy is not allowed. Message CKZ71010E is issued. |
| | | Other than active | Copy is not allowed. Message CKZ71010E is issued. |

Table 62. Copy outcomes depending on data sharing group member status and keyword values (continued)

| ALLOW-SKIP- INACTIVE-MEMBER value | Member included in DATA-SHARING- MEMBERS? | Member status | Result |
|---|---|-------------------|--|
| Y | Y | Active | Copy is allowed. |
| | | Other than active | Copy is allowed. |
| | N | Active | Copy is not allowed. Message CKZ71010E is issued. |
| | | Other than active | Copy is allowed. |

Target job changes

Make the following changes to your target job in order to apply Db2 logs.

- Ensure that the source subsystem Db2 SDSNEXIT library is in the STEPLIB for the target job.
- Add the following DDs to the target job. The data set names are required to be as shown.

```
//SYSINCKZ DD DISP=SHR,DSN=h1q.LOGAPCTL(membername)
//SYSOUT DD SYSOUT=*
//INFOM DD SYSOUT=*
```

- Optional: If you want to generate a summary report of the log apply process for the target job, add the following DD to the target job. The summary report includes information about the number of records that were applied to each object and the RBA/LRSN ranges that were used for each object.

```
//CKZLAREP DD SYSOUT=*
```

The summary report contains the following columns:

- DATABASE: Name of the source database.
 - OBJECT: Name of the source object (table space or index space).
 - PART: Partition number of the source object if partitioned; otherwise, this column contains 0000.
 - START RBA/LRSN: The starting RBA/LRSN of the first record that was applied to the target object.
 - END RBA/LRSN: The ending RBA/LRSN of the range of logs read. This not necessarily the last log record applied, but might be the last record that was read for the target job run.
 - RECS APPLIED: The number of records applied to the target object.
- If you are cloning across multiple LPARs, parameters used by the target job for TCP/IP communications to the source TCP/IP server job are passed to the target job from the source job using the SYNCDB2 data set. Include these parameters for the SET command:
 - SOURCE-TCPIP-SERVER-PORT - Connection port in the source TCP/IP server
 - SOURCE-TCPIP-STC-NAME - TCP/IP address space name in the target job
 - SOURCE-TCPIP-SERVER-IP - IP of the source TCP/IP server job

Chapter 20. Options for rebuilding indexes

When you clone table spaces, you might need to rebuild the indexes associated with the table spaces.

During the source job, Db2 Cloning Tool Table Space Cloning may exclude indexes from the cloning and mark them as requiring REBUILD for the following reasons:

- Source and target index comparison finds that there is an index or index partition mismatch. If the mismatch affects the entire index, then the entire index is excluded from the cloning and will be rebuilt in the target job. If the mismatch affects an index partition, then that index partition is excluded from the cloning and will be rebuilt in the target job.
- There is a version mismatch between the source index and the corresponding target index. The entire index is excluded from the cloning and will be rebuilt.
- There is an extension mismatch between the source index space and the corresponding target index space (the target has more extensions than the source). The entire index is excluded from the cloning and will be rebuilt.
- A problem is encountered during target data set preparation or the check for data set existence. The entire index is excluded from the cloning and will be rebuilt.
- DATA-MOVER PGM(SRCIMCPY) was specified, and the index COPY parameter value is NO. The entire index is excluded from the cloning and will be rebuilt.
- The source index is in PSRBD status (page set rebuild pending). The entire index is excluded from the cloning and will be rebuilt.
- The index was created on a column of a table on which DATA-MASKING is applied. The entire index is excluded from the cloning and will be rebuilt. For clone tables, if masks for both the base and clone tables are specified, both base and clone table indexes are excluded from the cloning and will be rebuilt.
- USE-LAST-CONSISTENT-FLASHCOPY(Y) is in use, the index is not included as part of the FlashCopy image copy, or the index and its table space are included in different FlashCopy image copies. Partitions and the base or clone instances of index spaces are excluded from the cloning and will be rebuilt (in accordance with messages CKZ54467I and CKZ54468I).

If RTS-COPY-ENABLED is set to Y, statistics are copied for these indexes to avoid problems with sorting during the rebuild process.

Indexes that are excluded from the cloning for rebuild in the source job are processed in the target job REBUILD template when a REBUILD job template is specified.

A report of excluded indexes is produced in the source job after the main COPY report. The report contains the following information:

```
CKZ55505I - Begin source excluded indexes report
CKZ55506I - The following indexes were not copied and will be rebuilt on target system

  DBNAME   ISNAME   TSNAME   EXCLUDE REASON   BASE/CLONE PART
  -----
DBNAME01  ISNAME01  TSNAME01  IX_PGFSIZE MISMATCH
DBNAME01  ISNAME04  TSNAME01  IX_PADDED MISMATCH
DBNAME02  ISNAME06  TSNAME02  IX_VERSION MISMATCH
DBNAME03  ISNAME03  TSNAME03  NOT IN FCIC
DBNAME04  ISNAME02  TSNAME04  NOT IN FCIC WITH TS           2
DBNAME05  ISNAME05  TSNAME05  IS EXTENT MISMATCH
DBNAME07  ISNAME07  TSNAME07  COPY NO                       C
DBNAME08  ISNAME08  TSNAME08  PSRBD STATUS                 BOTH      *
```

CKZ55507I - End source exclude indexes report

If REBUILD-INDEXES-EXECUTE is set to N, message CKZ54400I is produced to remind you that the INTELLIGENT-REBUILD DD should be specified in the target job. This DD allows Db2 Cloning Tool Table Space Cloning to generate the necessary REBUILD statements. Excluded indexes should be rebuilt to guarantee the availability of the target objects.

The REBUILD-INDEXES-SORTDEVT parameter allows you to specify the value of the SORTDEVT parameter for a REBUILD INDEX utility and specifies the device type for the temporary data sets that are to be dynamically allocated by the sort program. You can specify any disk device that is valid on the DYNALLOC parameter of the SORT or OPTION options for the sort program (although tape devices are not supported by the sort program). When rebuilding many large indexes is expected, you should specify a device type with low disk fragmentation for temporary data sets that are used by sort program.

There are several methods for rebuilding indexes from which to choose. The table that follows provides a brief overview of each method. Refer to the detailed topics on each method for specific information about setting up index rebuilds.

| Type | Description | Considerations | For more information |
|---------------------|---|---|--|
| Intelligent rebuild | Groups together as many indexes as possible into a single REBUILD utility for a given table space, to minimize table read I/O and elapsed time, as well as take advantage of built-in multi-tasking capabilities in the Db2 REBUILD utility. Runs the utilities at the end of the target job. | <ul style="list-style-type: none"> • More efficient when there are multiple indexes on tables. • Minimizes table read I/O and elapsed time by grouping indexes where possible. • Can report on and rebuild indexes that are only on the target. • If a CKZINTRB DD is supplied on the target job, a JCL version of the rebuild job stream is written to the specified file and can provide restart capability if the rebuild fails. • The REBUILD INDEX (ALL) statement is used if all of the indexes of the copied table space were not copied to the target. | “Method 1: Intelligent rebuild” on page 299 |
| Job templates | Job templates use JCL templates with variables to generate REBUILD JCL and write it to a specified file in the source or the target job. User intervention may be required before running the JCL. | Can provide restart capabilities, with manual intervention. | “Method 2: Using job templates in the source job” on page 300 or “Method 3: Using job templates in the target job” on page 300 |

Recommendations when rebuilding indexes

When rebuilding indexes as part of the cloning process, you may need to plan for restartability in case of rebuild failure. In addition, you might need to consider situations such as indexes that are only located on the source or only on the target, or whether you have partitioned objects that are split across multiple jobs.

Enabling restartability in the event of rebuild failure

An essentially identical JCL stream can be generated for potential use on a restart after a rebuild failure. Follow this procedure to restart the rebuild after a failure:

1. Determine the problem that caused the index rebuild job to fail.
2. Resolve the problem that caused the job to fail.
3. If necessary, terminate the failed utility ID by using a -TERM UTIL command.
4. Edit the JCL that was generated by either job templates or intelligent rebuild to remove the indexes that were successfully built.

- If JCL was generated by job templates, use the file specified by the *ddname*O DD in the target job (the *ddname* is taken from the source job keyword TARGET-JOB-INDEX-REBUILD-DDN(*ddname*)).
 - If the JCL was generated by intelligent rebuild, use the file specified by *ddname* CKZINTRB in the target job.
5. Submit the modified index rebuild job manually.

Excluding source-only indexes

To exclude source-only indexes, but copy all indexes that exist on both source and target, use ALWAYS-COPY-INDEXSPACES(Y) and exclude the source-only indexes using LISTDEF EXCLUDE.

Rebuilding target-only indexes

You can rebuild indexes that are only on the target by copying the related table space and specifying both PROCESS-UNMATCHED-TARGET-INDEXES(Y) and REBUILD-UNMATCHED-TARGET-INDEXES(Y) in the SET command of the source job.

Method 1: Intelligent rebuild

Intelligent rebuild groups together as many indexes as possible into a single REBUILD utility for a given table space. This minimizes table read I/O and elapsed time and takes advantage of built-in multi-tasking capabilities in the Db2 REBUILD utility. Intelligent rebuild runs the utility at the end of the target job.

About this task

To use this method, all index spaces can be explicitly included in the LISTDEF or can be included by specifying the COPY command ALWAYS-COPY-INDEXSPACES(Y). If the indexes of the copied table space exist on the source and are not copied to the target, then the REBUILD INDEX (ALL) statement is used to rebuild all indexes on the target.

INTELLIGENT-REBUILD is always enabled, regardless of the value that is specified in JCL. The behavior of INTELLIGENT-REBUILD is based on the value that is specified for REBUILD-INDEXES-EXECUTE and whether the CKZINTRB DD is provided, as follows:

- If REBUILD-INDEXES-EXECUTE(N) is specified and the CKZINTRB DD is specified in the target job, the statements for the REBUILD are generated and written to CKZINTRB DD, but not run.
- If REBUILD-INDEXES-EXECUTE(N) is specified and the CKZINTRB DD is not specified in the target job, the statements for the REBUILD are not generated, run, or written to any DD.
- If REBUILD-INDEXES-EXECUTE(Y) is specified and the CKZINTRB DD is specified in the target job, the statements for the REBUILD are generated, run, and written to the CKZINTRB DD.
- If REBUILD-INDEXES-EXECUTE(Y) is specified and the CKZINTRB DD is not specified in target job, the statements for the REBUILD are generated and run, but are not written to the CKZINTRB DD.
- If a REBUILD-INDEXES-SORTDEVT value is specified in the source job, the value of REBUILD-INDEXES-SORTDEVT is used for the value of SORTDEVT in all statements for the REBUILD that are generated by the intelligent rebuild process. If a REBUILD-INDEXES-SORTDEVT value is not specified in the source job, then SORTDEVT(SYSDA) is used in all statements for the REBUILD.

Procedure

1. If you want the REBUILD to be run at the end of the target job, specify SET command REBUILD-INDEXES-EXECUTE(Y) in the source job.
2. Specify SET command INTELLIGENT-REBUILD (Y) in the source job.
3. Optional: If you want to generate a JCL version of the INTELLIGENT-REBUILD job stream, include *ddname* CKZINTRB in the target job JCL. The JCL job can be used (with modifications) should a restart be required. The DD that is supplied must be LRECL=80 and RECFM=FB.

4. Optional: If you are cloning from image copies, indexes that are defined as COPY NO can be automatically rebuilt. Choose one of the following methods:
 - In the target job: Specify REBUILD-INDEXES-EXECUTE(Y) in the SET command of the source job.
 - In a generated REBUILD INDEX job: Specify either a CKZINTRB DD in the target job JCL, or specify the SET command parameter TARGET-JOB-INDEX-REBUILD-DDN in the source job.
5. Optional: Specify the device type for temporary data sets that are to be dynamically allocated by the sort program in the REBUILD-INDEXES-SORTDEVT keyword of the SET command in the source job.
6. Submit the source job.
7. Submit the target job. If REBUILD-INDEXES-EXECUTE(Y) is specified, the target job will dynamically rebuild the indexes.

Method 2: Using job templates in the source job

When you want to rebuild indexes that are not affected by LOG-APPLY, data masking, or cloning by using image copies, the indexes can be rebuilt from a utility job that is generated in the source job by using job templates.

About this task

This method uses job templates and the JOB-TEMPLATE command to rebuild indexes.

Note: This method may not produce the desired results if the indexes are not explicitly included.

Procedure

1. Specify the COPY command JOB-TEMPLATE (*inddname1*, *outddname1*) in the source job.
2. For *inddname1*, make a copy of one of the following sample templates in the SCKZJCL library and modify it for your site. Instructions for updating the template are contained in the sample templates.
 - CKZJOBI: This template can be used if you specified the COPY command ALWAYS-COPY-INDEXSPACES (Y) in the source job or explicitly specified the indexes by using LISTDEF. The template generates index rebuilds with REBUILD INDEXSPACE *database.indexspace* syntax.
 - CKZJOBR: This template can be used when the indexes are not explicitly included in the cloning, either by using LISTDEF or by using the COPY command keyword ALWAYS-COPY-INDEXSPACES(Y), in the source job. The template generates index rebuilds with REBUILD INDEX (ALL) TABLESPACE *database.table_space* syntax.
3. Submit the source job.

The utility JCL to rebuild the indexes is created and placed in *outddname1*, as specified in the JOB-TEMPLATE command.
4. To rebuild the indexes, submit the JCL in *outddname1*.

Method 3: Using job templates in the target job

When using LOG-APPLY or data masking, or when cloning from image copies, you can use job templates to create a job to rebuild the indexes. The job to rebuild the indexes is written to a member in the target job that you can submit.

About this task

This method uses job templates to rebuild indexes.

Note: This method may not produce the desired results if the indexes are not explicitly included.

Procedure

1. Specify the COPY command TARGET-JOB-INDEX-REBUILD-DDN(*ddname*) in the source job.

2. For *inddname1*, make a copy of one of the following sample templates in the SCKZJCL library and modify it for your site. Instructions for updating the template are contained in the sample templates.
 - CKZJOBI: This template can be used if you specify the COPY command ALWAYS-COPY-INDEXSPACES (Y) in the source job or explicitly specify the indexes by using LISTDEF. The template generates index rebuilds with REBUILD INDEXSPACE *database.indexspace* syntax.
 - CKZJOBR: This template can be used when the indexes are not explicitly included in the cloning, either by using LISTDEF or by using the COPY command keyword ALWAYS-COPY-INDEXSPACES(Y), in the source job. The template generates index rebuilds with REBUILD INDEX (ALL) TABLESPACE *database.table_space* syntax.
 - CKZJOB RN: This template can be used to generate partition-level rebuilds, using the automatic &PARTNUM variable. It can be used if you specify the COPY command ALWAYS-COPY-INDEXSPACES (Y) in the source job or explicitly specify the indexes by using LISTDEF. The template generates index rebuilds with REBUILD INDEXSPACE *database.indexspace* PART *partition_number* syntax.
3. Add the following two DDs to the target job for rebuilding indexes:

```
//ddnameI DD DISP=SHR,DSN=hlq.indsn(mbr)
```

where *ddname* is the DD that is specified in the TARGET-JOB-INDEX-REBUILD-DDN keyword. You must append an I to *ddname*. *hlq.indsn(mbr)* is the data set location and member name of the template that you edited. The default *ddname* value is CKZTIR, as provided in the Tools Customizer step that allocates table space cloning data sets (see [“Task: Application Cloning \(Table Space Cloning\) tasks”](#) on page 53).

```
//ddnameO DD DISP=OLD,DSN=hlq.outdsn(mbr)
```

where *ddname* is the DD that is specified in the TARGET-JOB-INDEX-REBUILD-DDN keyword. You must append an O to *ddname*. *hlq.outdsn(mbr)* is the data set location and member name where you want the utility job to be saved. The default *ddname* value is CKZTIR, as provided in the Tools Customizer step that allocates table space cloning data sets (see [“Task: Application Cloning \(Table Space Cloning\) tasks”](#) on page 53). This DD will contain the output from the generated REBUILD INDEX job.

4. Submit the source job.
5. Submit the target job. When the target job is run, the job to rebuild the indexes is written to *ddnameO*.
6. To rebuild the indexes, submit the JCL in *ddnameO*.

Chapter 21. Using job templates

Job templates allow you to generate jobs for data set copy utilities for source and target Db2 data sets, and jobs for some Db2 utilities for the table spaces and index spaces. These jobs are written to an output data set member and can then be submitted.

Job templates consist of the z/OS JCL statements, DSS commands, user and processing variables that Db2 Cloning Tool Table Space Cloning uses for input. You supply the job cards and the data sets needed by the utility and Db2 Cloning Tool Table Space Cloning supplies the names of the source and the target Db2 data sets or table spaces and index spaces. The job statements generated are then written to the output DD specified in the JOB-TEMPLATE subcommand.

Template variables should be separated from each other by a space, a comma, or a period. A template variable can be enclosed in parentheses or single quotation marks; for example, (&&VAR) or '&&VAR'. If you use parentheses, the opening parenthesis must be separated from the previous symbol with a space, comma, or period, and the closing parenthesis must be separated from the following symbol with a space, comma, or period. Each JCL line should contain only one predefined variable, except for &&DATABASE which must be followed by &&TABLESP, &&INDEXSP, or &&OBJECTSP. More than one user-defined variable (refer to the TEMPLATE-VARIABLE parameter in the SET command topic) may be presented on one JCL line in accordance with the rules of separation.

Refer to the sample jobs CKZJOB* that are provided in the SCKZJCL library. They contain detailed information about setting up a job template.

During the table space cloning process, job templates also can be used to generate utility jobs for target objects. Utility job templates are provided for CHECK DATA, CHECK INDEX, REBUILD INDEX, QUIESCE, REORG INDEXSPACE, REORG TABLESPACE, and RUNSTATS utilities.

The EMC API is supported using job templates. It is more restrictive in how the variables are used. Use sample job CKZJOB3 or CKZJOB8 (if the source objects are not stopped) to set up the EMC API template. In addition, this job can only be submitted after the source job has run with DATA-MOVER (PGM(NONE)).

For additional information about specific commands and parameters used for job templates, refer to the following:

JOB-TEMPLATE

This parameter provides the DD names that contain the templates. See the [“COPY” on page 513](#) command topic for more information.

CMDDNAME (DATA-MOVER subcommand)

This parameter provides the name of an output data set that contains a job built using the JOB-TEMPLATE. See the [“COPY” on page 513](#) command topic for more information.

TEMPLATE-VARIABLE

This parameter allows variables to be set for the life of the source job. See the [“SET” on page 567](#) command topic for more information.

Predefined processing variables

Predefined processing variables control how Db2 Cloning Tool Table Space Cloning generates the JCL. They are used to build DSS input command stream and job steps.

&&BEGDSSCC

Begin adding DSS continuation characters for data set list variables &SRCDNL, &TRGDNL and &PAIRDNL.

&&BEGDSSCM

(Required) Begin DSS commands. This should be the first variable after the SYSIN DD.

&&BEGDSSRC

(Required) Begin DSS repeat commands. This is the next variable after DSS commands that only execute one time per step. The DSS COPY command would be included after this variable.

&&BEGIDSPB

Start an individual data set processing block. Required for individual data set processing variables if there are cards without variables with the data set variables. For example, there might be an open parenthesis on a line by itself.

&&BEGPPAR

Add a parenthesis before the source and after the target data set when processing source target pairs. Required when using DSS RENAMEU, for example. Applies to data set list variables only, not individual data set processing variables.

&&BEGSTEP

(Required) Begin step generation. This must be before the z/OS EXEC JCL statement.

&&ENDDSSCC

End adding DSS continuation characters.

&&ENDDSSCM

(Required) End DSS commands.

&&ENDDSSRC

(Required) End DSS repeat commands.

&&ENDIDSPB

End an individual data set processing block.

&&ENDPPAR

Stop adding a parenthesis when processing source target pairs.

&&ENDSTEP

(Required) End step generation.

&&HEADER

Place this on a line at the top of the template on a line beginning with `//*`. It adds the date, time, job name and job number of the source job on that line. For example:

```
//* 08292 20:59:03.53 JOBNAME=PDDONAX8 JOBID=J0202482
```

&&PAIRDSNL

Data set pair (source and target) list. The character following this variable is the suffix character and will be added after each data set except the last target data set. No other characters can be on the line.

&&SRCDSN0, &&SRCDSN1, and &&SRCDSN2

Source data set. Each occurrence will output the next source data set. Use this variable along with `&&TRGDSN0`, `&&TRGDSN1`, or `&&TRGDSN2` to code the exact number and position of each data set in the DSS command string. The maximum number in one template is 255 as that is the maximum number of data sets DSS can process in a single COPY command. These variables provide more flexibility. They can, for example, be used to copy the target data sets back to the source data sets. `&&PAIRDSNL` only copies from the source to the target.

&&SRCDSNL

Source data set list. Each source data set will be output in the same order as the internal Db2 Cloning Tool Table Space Cloning copy command. The character following this variable is the suffix character and will be added after each data set except the last. No other characters can be on the line.

&&SRCOBS

Source objects.

&&TRGOBS

Target objects.

&&SRCSSID

Source Db2 SSID.

&&STEPNUM

Add this on the step execution card to request a three-digit number appended to the step name. For example:

```
//STEP.&&STEPNUM EXEC PGM=ADDRSSU,REGION=4M
```

Note: The dot between STEP and &&STEPNUM indicates the variable value is to be concatenated with the preceding string.

&&TRGDSN0, &&TRGDSN1, and &&TRGDSN2

Target data set. Each occurrence will output the next target DSN. Use this variable along with &&SRCDSN0, &&SRCDSN1, and &&SRCDSN2 to code the exact number and position of each data set in the DSS command string. The maximum number in one template is 255 as that is the maximum number of data sets DSS can process in a single COPY command.

&&TRGDSNL

Target data set list. Each target data set will be output in the same order as the internal Db2 Cloning Tool Table Space Cloning copy command. The character following this variable is the suffix char and will be added after each DSN except the last. No other characters can be on the line.

&&TRGSSID

Target Db2 SSID.

&&TRGVCAT

Target VCAT (DEFVCAT in the TARGET-DB2 command).

Data set processing variables

There are seven data set processing variables. Each must begin on a separate line and may be followed by one character, a comma or a close parenthesis used to terminate the DSS data set subcommand.

&&SRCDSNL, &&TRGDSNL, &&PAIRDSNL

These variables use the PARMLIB member value of DSNS_PER_COPY to determine how many data sets are included with each DSS COPY command.

&&SRCDSN0, &&SRCDSN1, &&SRCDSN2, &&TRGDSN0, &&TRGDSN1, &&TRGDSN2

These variables determine the number of data sets per COPY command by the number of variables found. For example, if the parameter value for the number of data sets per DSS COPY command is 255 and 50 &&SRCDSN variables are found, 255 is reduced to 50 while this template is being processed. It is changed back to 255 for the next template (if any). If more than 255 &&SRCDSN or &&TRGDSN variables are found, the job ends with an error.

Block processing variables

There are three sets of required block processing variables: z/OS JCL step, DSS commands and DSS repeat commands.

The block processing variables must be present in every template and must be in the correct order. They are in order of appearance in the template:

- &&BEGSTEP
- &&BEGDSSCM
- &&BEGDSSRC
- &&ENDDSSRC
- &&ENDDSSCM
- &&ENDSTEP

Other block processing variables control DSS continuation characters and when to enclose data sets in parentheses. They are:

- &&BEGDSSCC

- &&BEGIDSPB
- &&BEGPPAR
- &&ENDDSSCC
- &&ENDIDSPB
- &&ENDPPAR

Job templates for utilities

During the table space cloning process, job templates can be used to generate utility jobs for target objects. Utility job templates are provided for CHECK DATA, CHECK INDEX, REBUILD INDEX, QUIESCE, REORG INDEXSPACE, REORG TABLESPACE, REPAIR, and RUNSTATS utilities.

The utility job templates are located in the SCKZJCL library. The following table describes the utility jobs and their member names.

| Db2 utility | Member name |
|--|-------------------------------|
| CHECK DATA | CKZJOB CD |
| CHECK INDEX | CKZJOB CI |
| QUIESCE | CKZJOB Q |
| REBUILD INDEX | CKZJOB I, CKZJOB R, CKZJOB RN |
| REORG INDEXSPACE | CKZJOB RI |
| REORG TABLESPACE | CKZJOB RO |
| REORG TABLESPACE (Db2 V11 only, when converting from 6-byte to 10-byte RBA/LRSN) | CKZJOB RX |
| REPAIR | CKZJOB RP |
| RUNSTATS | CKZJOB RS |

Use the templates as follows:

1. Edit the utility template by following the instructions in the member. Generally, you will add a job card, and modify the job's user ID and data set high level qualifiers.
2. In the source job JCL, add an input DD and data set name to point to the edited utility job template.
3. In the source job, add an output DD and data set name. This library member will contain the generated utility job that you need to submit after the source and target jobs have successfully run.
4. Submit the source job.
5. Submit the target job.
6. Verify that the target objects have been successfully created on the target subsystem.
7. Submit the utility job.

Variables in utility job templates

The variables used in the utility job templates are as follows:

&&JOB CARD

This variable is used to indicate the end of job card information.

&&DATABASE

This variable is used to insert a database name in the utility command.

&&TABLESP

This variable is used to insert a table space name in the utility command.

&&OBJECTTY

This variable is used to insert TABLESPACE or INDEXSPACE in the utility command. This variable allow you to use the same template for both table spaces and index spaces. Refer to the example that follows for more information.

&&OBJECTSP

This variable is used to insert a table space or index space name in the utility command. This variable allow you to use the same template for both table spaces and index spaces. Refer to the example that follows for more information.

&&INDEXSP

This variable is used to insert an indexspace name in the utility command.

&&PARTNUM

This variable inserts a partition number when an object is partitioned. When &&PARTNUM is specified, a utility command line is written for each partition.

&&TRGSSID

This variable defines the target subsystem on which the utility will be run.

&&TRGOBJS

This variable defines the target objects on which the utility will be run.

&&INCRVAL

This variable is used as an incremental in constructing data set names used during the utility, such as SYSREC and SYSUT1.

&&UTLRBIX

This variable is used to identify that a template contains a REBUILD utility and to select the objects that are required for the REBUILD. This variable is specified in comments.

&&UTLREP

This variable is used to identify that a template contains a REPAIR utility and to select the objects that are required for the REPAIR. This variable is specified in comments.

The &&OBJECTTY and &&OBJECTSP variables should only be specified together. Using the &&OBJECTTY and &&OBJECTSP variables allows both table spaces and index spaces to be processed in the same job template. For example, in a REORG utility template, you can either:

- Create two separate templates, one for table spaces and one for index spaces, with REORG syntax as follows:

```
REORG
TABLESPACE &&DATABASE.&&TABLESP
SHRLEVEL REFERENCE
```

```
REORG
INDEXSPACE &&DATABASE.&&INDEXSP
SHRLEVEL REFERENCE
```

- Or, you can create one template with the following REORG syntax:

```
REORG &&OBJECTTY
&&DATABASE.&&OBJECTSP
SHRLEVEL REFERENCE
```

Job templates for cloning consistent FlashCopy data sets to Db2 data sets using IBM Sterling Connect:Direct

Job templates can be used to clone consistent FlashCopy data sets from the source system to Db2 data sets on the target system using Sterling Connect:Direct. When using the provided templates, the source

and the target systems do not need to share DASD and the cloning process does not require interim volumes.

Before you begin

The job templates for this cloning process are located in the SCKZJCL library. The following table describes the templates.

| Table 65. Job template names and purpose | |
|--|--|
| Template name | Purpose |
| CKZJOBT4 | Cloning using Sterling Direct:Connect |
| CKZJOBT5 | Optional template that adds volume candidates to the target data sets before cloning using Sterling Direct:Connect |

About this task

These steps describe manual set up and generation of the job templates.

Procedure

1. Modify CKZJOBT4 to add a job card and to set the variables as described in the template. The template contains information as to how the variables must be set.

Note: You should review the GENPROC step in CKZJOBT4. This step generates the Sterling Connect:Direct process to copy the Db2 data sets. One of its steps preallocates nonexistent target data sets to ensure that they have the correct VSAM attributes. You might want to change the number of automatically added candidate volumes, or the primary and secondary space quantities.

2. Optional: Modify the CKZJOBT5 template to add a job card and add volume candidates that can be used for the target data sets.
3. Modify the source job as follows:

- Add the following DDs to include the CKZJOBT4 template:

```
CKZSCDI DD DISP=SHR,DSN=h1q1.SCKZJCL(CKZJOBT4)
CKZSCDO DD DISP=SHR,DSN=h1q2.JCLLIB(SCDJOB)
```

- **Optional:** Add the following DDs if you plan to use the CKZJOBT5 template to add candidate volumes:

```
CKZALTDI DD DISP=SHR,DSN=h1q1.SCKZJCL(CKZJOBT5)
CKZALTD0 DD DISP=SHR,DSN=h1q2.JCLLIB(ALTJOB)
```

- Add the following variables to the SET command:

```
TEMPLATE-VARIABLE ( -
    &&PNODE,pnode_name, -
    &&SNODE,snode_name, -
    &&SRCTJDSN,source_target_job_data_set_name, -
    &&TRGTJDSN,target_target_job_data_set_name, -
    &&SRCHLQ,source_syncdb2_hlq, -
    &&TRGHLQ,target_syncdb2_hlq, -
    &&SCDHLQ=Connect:Direct_hlq_or_alias) -
```

- Add the following DDs to the JOB-TEMPLATE keyword of the COPY command:

```
JOB-TEMPLATE(
    CKZALTDI, CKZALTD0,
    CKZSCDI, CKZSCDO
)
DSS-COPY-COMMANDS(255) -
```

Note: CKZALTDI and CKZALTD0 are only required if you plan to run the CKZJOBT5 job.

- Ensure that the following parameters are set as shown to clone data from consistent FlashCopy image copies:

```
DATA-MOVER(
  PGM(NONE)
  USE-LAST-CONSISTENT-FLASHCOPY(Y)
)
```

4. Run the source job. The source job generates jobs in *hlq.TDDOUT(CKZJOB4)* and *hlq.TDDOUT(CKZJOB5)*.
5. Optional: To add volume candidates to the target subsystem, run the CKZJOB5 job.
6. Submit the *hlq.TDDOUT(CKZJOB4)* job to run the Sterling Connect:Direct copies.
7. Run the target job.

Using the ISPF interface and job templates to clone with IBM Sterling Connect:Direct

You can use the ISPF interface to generate the jobs for cloning with Sterling Direct:Connect.

Procedure

1. In the SET command for the source job, specify the job template variables on the **Specify Job Template Variables** panel.

For example:

```

                                Specify Job Template Variables
Command ==>>                                Scroll ==>> CSR

Commands:      N - New Line  A - Select All  C - Clear All Selections
Line commands: D - Delete Line  S - Select/Unselect

Creator . . . : TWUSR           Name . . . . : TEST-4698
Share Option . : UPDATE         Description . :
Source SSID .  : SS01           Target SSID . : SS02

                                Row 1 of 7
-----
Name      Value
-----
SEL PNODE AB1S
SEL SNODE AB2T
SEL SRCTJDSN USER.JCLLIB
SEL TRGTJSDN USER.JCLLIB.TARGET
SEL SRCHLQ  USER.CKZ32
SEL TRGHLQ  USER.CKZ32.TARGET
SEL SCDHLQ  $CDVSAM
***** Bottom of data *****

```

2. In the COPY command for the source job, specify the job template DDs on the **Specify Job Template Data Sets and Members** panel.

For example:

```

Specify Job Template Data Sets and Members
Command ==> Scroll ==> CSR

Commands:      A - Add Line
Line commands: D - Delete Line

Creator . . . : TWUSR          Name . . . . : TEST-4698
Share Option . : UPDATE       Description . :
Source SSID .  : SS01         Target SSID . : SS02

TEMPLATE INPUT DATA SET . . CKZ32.SCKZJCL
TEMPLATE OUTPUT DATA SET . . USER.SCD.JCLLIB

Row 1 of 2

Cmd INPUT MEMBER INPUT DDNAME OUTPUT MEMBER OUTPUT DDNAME
-----
      CKZJOB5    CKZALTDI   ALTJOB      CKZALTD0
      CKZJOB4    CKZSCDI   SCDJOB      CKZSCD0
***** Bottom of data *****

```

3. In the COPY command for the source job, ensure that DATA-MOVER PGM(NONE) and USE-LAST-CONSISTENT-FLASHCOPY(Y) are set.
4. Generate the source jobs using the ISPF UI.
The source job generates jobs in *hlq*.TDDOUT(CKZJOB4) and *hlq*.TDDOUT(CKZJOB5).
5. Optional: To add volume candidates to the target subsystem, run the CKZJOB5 job.
6. Submit the *hlq*.TDDOUT(CKZJOB4) job to run the Sterling Connect:Direct copies.
7. Run the target job.

Cloning source VSAM files using Sterling Connect:Direct

You can clone VSAM data sets from the source system to Db2 data sets on the target system using Sterling Connect:Direct with the CKZJOB4 and CKZJOB5 templates.

Important: To avoid consistency problems and for operational reasons, the source objects must be stopped before running the CKZJOB4 job. Sterling Connect:Direct requires that the data set is not allocated to another task.

To clone source VSAM data sets using the templates, use the steps in “Job templates for cloning consistent FlashCopy data sets to Db2 data sets using IBM Sterling Connect:Direct” on page 307 with the following additional instructions.

- In the COPY command, specify USE-LAST-CONSISTENT-FLASHCOPY(N) with DATA-MOVER PGM(NONE). The value of FUZZY-COPY does not affect these jobs; you can specify either FUZZY-COPY(Y) or FUZZY-COPY(N).
- Before running the CKZJOB4 and/or CKZJOB5 jobs, stop the source objects using the job that is generated for stopping the source objects. The target objects should also be stopped.
- When you run the source job, the TCP/IP server job also should be run.
- Run the source job to generate the Connect:Direct copy job on the source LPAR.
- Run the Connect:Direct generated copy job to move the source data sets to the target location with target names. The source objects must be stopped during the copying of the VSAM data sets. Log apply is not required, but if you are cloning XML objects, then you should use the source TCP/IP server job.

Note: The Connect:Direct template includes cloning of data sets with data and the SYNCDB2 data sets. The Connect:Direct template currently does not clone real-time statistics data sets.

- Run the target job on the target LPAR. When you run the target job, the source TCP/IP server job also should be run if you are cloning XML objects, or will use log apply functionality.

Chapter 22. Using the ISPF interface

The new Db2 Cloning Tool ISPF interface allows you to clone table spaces and index spaces that are required for your applications using task-driven ISPF panels. Sources for application cloning (formerly called table space cloning) can be Db2 data sets, image copies, or user copies. With the new interface, you can interactively select the objects that you want to clone. You can also clone applications across subsystems without shared DASD.

This topic describes how to use the new ISPF interface that is available in Db2 Cloning Tool V3.2 after applying PTF PH09578. For information about using the classic ISPF interface that is installed prior to PH09578, refer [Chapter 23, “Using the classic ISPF interface,”](#) on page 323.

Features

The new ISPF interface facilitates application cloning with the following features and enhancements:

- Provides a more intuitive, task-driven interface for creating application cloning jobs. You do not need to be familiar with table space cloning commands in order to use the interface.
- Online interactive selection of source objects for application cloning. You can generate lists of databases, table spaces, tables, indexes, or storage groups, and select the objects from real-time lists. This functionality requires a Db2 connection; however, any Db2 connection can be used.
- You can create application cloning profiles so you can re-use and repeat cloning processes, but cloning can also be accomplished without creating a profile.
- Existing table space cloning profiles can be migrated into application cloning profiles for use in the new interface.

Getting started with the ISPF interface

After product customization is complete, follow these steps to start using the new ISPF interface.

Before you begin

The Db2 Cloning Tool ISPF interface requires a minimum region size of 50000 KB.

Note: Before attempting to use the ISPF interface, ensure that customization steps been completed as described in [Chapter 4, “Customizing Db2 Cloning Tool,”](#) on page 77 and [Chapter 5, “After customizing Db2 Cloning Tool using Tools Customizer,”](#) on page 97.

If you have existing table space cloning profiles that you want to use in the new ISPF interface, ensure that the steps in the topic [“Migrating table space cloning profiles to application cloning profiles”](#) on page 103 have been completed.

Every product panel in Db2 Cloning Tool has its own help panel. The help panel lists the purpose of the panel, available commands, and fields and column data that is displayed. Enter HELP or press PF1 to display a help panel. Detailed information about command parameters and other valid values on the panels are provided in the help panels. The panel values are not described in detail in these topics.

Procedure

1. Run the CKZ CLIST. The **Db2 Cloning Tool for z/OS Primary Option Menu** is displayed, as shown in the figure.

Figure 12. Primary Option Menu

```

Db2 Cloning Tool for z/OS      Primary Option Menu
Option ==>

0  User Settings                                User ID . . : TWUSR
                                           System ID . . : LPAR
Application Clone                               Appl ID . . : CKZ
1  Manage Application cloning profiles          Version . . : 3.2
2  From Db2 source data sets
3  From image copies
4  Using a user copy
5  Without shared DASD

Subsystem Clone
6  Manage Subsystem cloning profiles

9  Use classic ISPF interface
X  Exit

Db2 control data set . . : PDUSERS.CKZ.QA.CONTROL
Repository HLQ . . . . : PSUSERS.CKZ320.QA

```

2. First time users: Enter 0 in the **Option** line to set up job cards and Db2 source and target subsystem information.
 - a) On the **User Settings** panel, enter 0 to provide job card information.
See the topic [“Configuring job cards”](#) on page 312 for more information.
When finished, press PF3 (END).
 - b) On the **User Settings** panel, enter 1 to set up the Db2 subsystems that you will be using for cloning.
 - c) On the **Db2 Subsystems** panel, create one or more Db2 subsystems that will be source and targets for cloning.
See the topic [“Configuring Db2 subsystems”](#) on page 312 for more information.
When finished, press PF3 (END) until the **Primary Option Menu** is displayed.
3. Do one of the following:
 - To build application cloning jobs, enter one of the options 1 through 5 on the **Primary Option Menu**.
 - To build subsystem cloning jobs, enter option 6 or 9 on the **Primary Option Menu**. This option uses the classic ISPF interface.
 - To build table space cloning jobs using the classic ISPF interface, enter option 9 on the **Primary Option Menu**.

Configuring job cards

Job card information that will be used for all cloning jobs must be provided.

Procedure

1. On the **Primary Option Menu**, enter 0 in the **Option** line and press Enter.
2. On the **User Settings Menu**, enter 0 in the **Option** line and press Enter.
3. On the **Set Batch Job Card Information** panel, enter your job card in the **Job Card Information** area.
4. When finished, press PF3 (END).

Configuring Db2 subsystems

Db2 subsystem information for all SSIDs that will be used as source and target subsystems, or for DDF connections to a source subsystem, must be provided before application cloning jobs can be built.

Procedure

1. On the **Primary Option Menu**, enter 0 in the **Option** line and press Enter.

2. On the **User Settings** panel , enter 1 in the **Option** line and press Enter.
3. On the **Db2 Subsystems** panel, enter C in the **Command** field and press Enter.
4. On the **New Db2 Subsystem** window, enter the Db2 SSID of the subsystem to add and press Enter.
5. On the **Db2 Subsystem Parameters** panel, verify the product plan name to be used when connecting to the Db2 catalog.
The plan name was provided during product customization.
6. Provide values for your subsystem in the following fields:
 - **Db2 ZPARMs member**
 - Both **Db2 Bootstrap DSN** fields
 - At least one **Db2 Loadlib** field.
7. Select the option to define subsystem information that is required for each type of cloning.
 - a) Enter 1 to specify subsystem cloning information.
When you press Enter, the **Subsystem Cloning Information** panel is displayed; refer to the help panel for information about the fields on this panel.
 - b) Enter 2 to specify application cloning information.
When you press Enter, the **Application Cloning Information** panel is displayed; refer to the help panel for information about the fields on this panel.
8. When finished, press PF3 (END) until you return to the **Primary Option Menu**.

Working with application cloning profiles

Application cloning profiles are used to save cloning parameters and job settings for application cloning scenarios. Application cloning profiles are used to build all of the required jobs to clone the table spaces and indexes for an application. Each application cloning type has its own profile type.

The new ISPF interface does not require that you create application cloning profiles to build cloning jobs. However, the cloning parameters for those jobs are not saved until you save them as a profile. You can set up cloning parameters first, and create a profile after completing the settings, as long as you do not exit the **Application Cloning** panel.

Although application cloning profiles are saved in the same VSAM repository as table space cloning profiles, they are not compatible with the classic ISPF interface that is available in Db2 Cloning Tool prior to PTF PH09578.

Creating an application cloning profile while setting cloning parameters

You can also create an application profile by saving the cloning profile parameters that you have entered as a profile.

Procedure

1. On the **Application Cloning** panel for the specified type of cloning, enter a profile name in the **Name** field, and optionally a description in the **Description** field. These fields are shown highlighted in the following figure:

```

Application Cloning From Db2 Source Data Sets
Command ==>>

Use this panel to clone data from the source to the target Db2 data sets.

Commands: S - Set source information      T - Set target information
          C - Set connection options      L - Set log apply options
          A - Set advanced options        P - Save options as a profile
          B - Build cloning JCL jobs

Creator . . . . . TWDEB      Name . . . . .
Share Option . . UPDATE      Description . .
                                     More:      >
                                     +
Source SSID . . . . . DC1Q    (? to select from list)
  Access Source through SSID (DDF) . . DC1Q    (? to select from list)
Target SSID . . . . . DC1A    (? to select from list)

Source availability during cloning . . YES    (Yes/No)
Target consistency (log apply) . . . . YES    (Yes/No)
Clone statistics . . . . . YES    (Yes/No)

Processing options:
  Stop job on warnings . . . . . NO    (Yes/No)
  Copy data options

```

Figure 13. Profile creation fields

- Optional: Modify the profile **Creator** and **Share Option** fields. The defaults that are provided are the current user ID for the profile creator and UPDATE for the share option. Refer to the help panel for descriptions of these fields and their possible values.
- To save the profile, press PF3 (END). The new profile is listed on the Manage Application Cloning Profiles panel CKZ1MACP, not shown.
- Enter P in the Command field and press Enter.
The profile is saved.

What to do next

You can build, edit, and delete saved profiles as described in the topic [“Managing application cloning profiles”](#) on page 315.

Creating an application cloning profile while viewing the profile list

You can create an application profile before setting up the cloning profile parameters when you are viewing the list of profiles on the **Manage Application Cloning Profiles** panel.

Procedure

- On the **Primary Option Menu**, enter 1 in the **Option** line and press Enter.
- Optional: On the **Enter Application Profile Selection Criteria** window, enter selection criteria in the Profile Like or Creator Like field, and enter a value in the Clone Method field and press Enter.
If entered, the values in these fields limit the display of profile to specific names, creators, or type of profiles.
When finished, press Enter.
- On the **Manage Application Cloning Profiles** panel, enter C in the **Command** field and press Enter.
- On the **Create Application Cloning Profile** panel, enter the value for the type of application cloning that you want to perform and press Enter.
The **Application Cloning** panel for the type of cloning that you specified is displayed.
- Enter cloning parameters, then save the profile as described in [“Creating an application cloning profile while setting cloning parameters”](#) on page 313.

Managing application cloning profiles

From the application cloning profile list, you can edit, view, copy and rename the profiles, as well as build cloning jobs.

Procedure

1. On the **Primary Option Menu**, enter 1 in the **Option** line and press Enter.
2. On the **Enter Application Profile Selection Criteria** window, enter selection criteria in the Profile Like or Creator Like field, and enter a value in the Clone Method field and press Enter.
If entered, the values in these fields limit the display of profile to specific names, creators, or type of profiles.
When finished, press Enter. The **Manage Application Cloning Profiles** panel lists the profiles that meet your selection criteria.
3. Use the line commands that are listed on the panel to browse, delete, view, rename, and copy existing application cloning profiles, and build application cloning jobs.

Application cloning from Db2 source data sets

Db2 Cloning Tool copies application data from the source Db2 data sets to the target Db2 data sets using the DFSMS ADRDSSU COPY utility. Additional processing of the target objects is performed by the Db2 Cloning Tool internal processor and by Db2 utilities.

Procedure

1. On the **Application Cloning From Db2 Source Data Sets** panel, enter or select the source and the target SSIDs.
For more information, see [“Selecting the source and target Db2 SSIDs” on page 318](#).
2. Enter **S** in the **Command** field to specify the source objects to be cloned.
For more information, see [“Selecting source objects” on page 318](#).
3. In the **Source availability during cloning** field, specify whether the source must be available to users during cloning. If you enter YES, source objects are not stopped or started during the cloning. If you enter NO, source objects are started and stopped during cloning.
4. In the **Target consistency (log apply)** field, specify whether you want the target to be consistent after the cloning.
If the **Source availability during cloning** field is set to YES, to ensure data consistency, you should set **Target consistency (log apply)** field to YES.
5. Set options under the **Processing options** heading as desired, or accept the defaults.
6. Set options under the **Performance settings** heading as desired, or accept the defaults.
7. Set values for processes that are run on the target subsystem by entering T in the **Command** field.
These processes include object translation specifications and whether DDL is to be generated to create non-existent or re-create existing target objects.
8. If **Target consistency (log apply)** is set to YES, enter L in the **Command** field to set log apply options.
9. To set the type of copy utility that will clone the objects, in addition to object mismatch processing options and other advanced cloning options, enter **A** in the **Command** field.
10. Optional: When finished setting cloning options, save the settings as a profile by entering P in the **Command** field.
11. To build the cloning jobs, enter B in the **Command** field.
For more information, see [“Building application cloning jobs” on page 319](#).
12. Run the cloning jobs.
For more information, see [“Submitting jobs for application cloning from Db2 source data sets” on page 321](#).

Application cloning from image copies

Db2 Cloning Tool copies application data from source Db2 image copies to the target Db2 data sets. The source can be the last consistent FlashCopy image copy, or can be a specific point in time used to select the image copy.

Procedure

1. On the **Application Cloning From Image Copies** panel, enter or select the source and the target SSIDs.
For more information, see [“Selecting the source and target Db2 SSIDs” on page 318.](#)
2. Enter **S** in the **Command** field to specify the source objects to be cloned.
For more information, see [“Selecting source objects” on page 318.](#)
3. In the **Image Copy Processing options** field, specify which image copy type you want to use as the source for the cloning.
 - If you enter 1, the last consistent FlashCopy image copy is used.
 - If you enter 2, Db2 image copies are used. You must also specify the point up to which you want to process the image copies in the **Point in time** field.
4. If you selected Db2 image copies as the source, in the **Target consistency** field, indicate whether you want the target to be consistent after the cloning.
Note: If you specify NO, data on the target might be inconsistent with the source if the source objects are in read/write status. However, if you want the same data on the target that is in the source object image copies, specify NO.
5. Set options under the **Processing options** heading as desired, or accept the defaults.
6. Set options under the **Performance settings** heading as desired, or accept the defaults.
7. Set values for processes that are run on the target subsystem by entering T in the **Command** field.
These processes include object translation specifications and whether DDL is to be generated to create non-existent or re-create existing target objects.
8. If **Target consistency** is set to YES, enter L in the **Command** field to set log apply options.
9. To set the type of copy utility that will clone the objects, in addition to object mismatch processing options and other advanced cloning options, enter **A** in the **Command** field.
10. Optional: When finished setting cloning options, save the settings as a profile by entering P in the **Command** field.
11. To build the cloning jobs, enter B in the **Command** field.
For more information, see [“Building application cloning jobs” on page 319.](#)
12. Run the cloning jobs.
For more information, see [“Submitting jobs for application cloning from image copies” on page 321.](#)

Application cloning using user copies

Db2 Cloning Tool can clone application data from the source Db2 data sets to the target Db2 data sets using user copies.

About this task

In this scenario, source objects are not automatically stopped or started. If you want to stop the source objects, use the JCL job that is generated by the source job.

Procedure

1. On the **Application Cloning Using a User Copy** panel, enter or select the source and the target SSIDs.

- For more information, see [“Selecting the source and target Db2 SSIDs” on page 318.](#)
2. Enter **S** in the **Command** field to specify the source objects to be cloned.
For more information, see [“Selecting source objects” on page 318.](#)
 3. In the **Source data set processing options** field, specify the source of the data that you want to clone.
 - If you enter 1, the last consistent FlashCopy image copy is used.
 - If you enter 2, Db2 source data sets are used.
 4. If you specified Db2 source data sets as the source, in the **Target consistency** field, specify whether you want the target to be consistent after the cloning.
Note: If you specify YES, the UNLOAD/LOAD feature cannot be used to resolve mismatches and clone data.
 5. Set options under the **Processing options** heading as desired, or accept the defaults.
 6. Set options under the **Performance settings** heading as desired, or accept the defaults.
 7. Set values for processes that are run on the target subsystem by entering T in the **Command** field.
These processes include object translation specifications and whether DDL is to be generated to create non-existent or re-create existing target objects.
 8. If **Target consistency** is set to YES, enter L in the **Command** field to set log apply options.
 9. To set object mismatch processing options and other advanced cloning options, enter **A** in the **Command** field.
 10. Optional: When finished setting cloning options, save the settings as a profile by entering P in the **Command** field.
 11. To build the cloning jobs, enter B in the **Command** field.
For more information, see [“Building application cloning jobs” on page 319.](#)
 12. Run the cloning jobs.
For more information, see [“Submitting jobs for application cloning from user copies” on page 321.](#)

Application cloning without shared DASD

When no DASD is shared between the source objects and the targets objects, application data can be cloned by using the Sterling Connect:Direct utility.

About this task

This cloning process builds jobs to copy the target job, the required SYNCDB2 data set, the source Db2 data sets, and real time statistic data sets (if specified) from the source to target LPAR.

Procedure

1. On the **Application Cloning without Shared DASD** panel, enter or select the source and the target SSIDs.
For more information, see [“Selecting the source and target Db2 SSIDs” on page 318.](#)
2. Enter **S** in the **Command** field to specify the source objects to be cloned.
For more information, see [“Selecting source objects” on page 318.](#)
3. In the **Source data set processing options** field, specify the source of the data that you want to clone.
 - If you enter 1, the last consistent FlashCopy image copy is used. Source objects will not be stopped.
 - If you enter 2, Db2 source data sets are used. To select this option, the **Source availability during cloning** field must be set to NO.
4. Set options under the **Processing options** heading as desired, or accept the defaults.
5. Set options under the **Performance settings** heading as desired, or accept the defaults.

6. Set values for processes that are run on the target subsystem by entering T in the **Command** field. These processes include object translation specifications and whether DDL is to be generated to create non-existent or re-create existing target objects.
7. To set Connect:Direct job options, object mismatch processing options, and other advanced cloning options, enter **A** in the **Command** field.
8. Optional: When finished setting cloning options, save the settings as a profile by entering P in the **Command** field.
9. To build the cloning jobs, enter B in the **Command** field.
For more information, see [“Building application cloning jobs” on page 319](#).
10. Run the cloning jobs.
For more information, see [“Submitting jobs for application cloning without shared DASD” on page 322](#).

Selecting the source and target Db2 SSIDs

Source and target Db2 SSIDs must be specified for all application cloning profile types.

Before you begin

Any Db2 subsystem that will be used in application cloning must first be configured using option 0.1 on the **Primary Option Menu**.

Procedure

- On the Application Cloning panel for the specified type of cloning, do one of the following:
 - Enter the source and target Db2 SSID in the **Source SSID, Access Source through SSID (DDF), and Target SSID** field.
 - Enter a ? in any of those fields to select the subsystem from the list of configured Db2 subsystems.

The **Access Source through SSID (DDF)** field can be specified when the source subsystem might not be available to connect to when accessing source objects. Refer to the help panel for additional information.

Selecting source objects

Source objects for application cloning can be selected from an interactive list, or can be imported from an existing LISTDEF member that contains the list of source objects to be cloned.

Selecting source objects from a list

You can use an interactive set of panels to retrieve the source objects for application cloning.

Procedure

1. On the **Application Cloning** panel for the specified type of cloning, enter S in the **Command** field and press Enter.
2. On the **Add Objects** window, enter 1 to add the object list using the interactive panels.
3. On the **Object Types Selection** panel, enter S next to the type of objects from which you want to select and press Enter.
4. A filter window for the type of object that you selected displays.
Specify selection criteria in the **Name Like** field, whether you want the objects to be wildcarded, and whether the selected objects are to be included or excluded from the cloning. Additional fields might be available for filtering, depending on the object type. When finished, press Enter.
A selection panel for the type of object that you selected displays.

5. Enter S next to the objects that you want to select for cloning and press Enter.
6. If you selected multiple types of objects, the filter window and the selection repeats for each type of object.

When all object types have been selected, the **Source Object List** is displayed.

7. Review and modify the list of source objects as needed.

Several columns on the right side of the panel allow you to specify the type of related objects to be included with a particular object (RI-related objects, related indexes, LOB and XML spaces, and history tables). Scroll right to see all the columns. Refer to the help panel for additional information about these columns.

8. When finished, press PF3 (END) to save the list.

Importing a source object LISTDEF

You can import a LISTDEF that contains the list of source objects to be cloned into the application cloning source object list.

Before you begin

The LISTDEF to be imported must be a member in a partitioned data set and must contain standard Db2 LISTDEF syntax.

Procedure

1. On the **Application Cloning** panel for the specified type of cloning, enter S in the **Command** field and press Enter.
2. On the **Add Objects** window, enter 2 to import the object list from a LISTDEF data set.
3. On the **Import Options** panel, enter the data set name and member name of the LISTDEF member to import and press Enter.
The **Source Object List** is displayed.
4. Review and modify the list of source objects as needed.
Several columns on the right side of the panel allow you to specify the type of related objects to be included with a particular object (RI-related objects, related indexes, LOB and XML spaces, and history tables). Scroll right to see all the columns.
5. When finished, press PF3 (END) to save the list.

Building application cloning jobs

After you set the cloning parameters, build the JCL jobs. You can build jobs when viewing an application cloning profile list by entering the B line command next to the profile. You can also build jobs by entering the B primary command on the **Application Cloning** panel for the type of cloning that you are performing.

Procedure

1. Enter B next to an application cloning profile, or in the **Command** field on the **Application Cloning From Db2 Source Data Sets** panel.
The **Build Application Clone Jobs** panel is displayed.
2. If you want to run the jobs in simulate mode, enter YES in the **Use simulate mode** field. In simulate mode, Db2 Cloning Tool verifies syntax and the source objects, and generates a list of source objects in a report. In addition, Db2 Cloning Tool checks for and reports on object mismatches, and allocates and verifies target objects and data sets. No copies are performed in simulate mode.
3. Enter the data set and member names for each cloning job in the fields under the **The jobs to do the Application Clone will be placed in the provided data set:** heading, or accept the defaults. Only jobs that are required for the cloning have input fields.

Note that this area is scrollable; place your cursor inside the area and press the PF8(DOWN) and PF7(UP) keys.

4. Verify the job processing options, or accept the defaults.
5. Verify the job card information.
6. Press PF3 (END).
7. On the **DD Specifications for Source Job** panel, verify the DDs for the source job.
8. Review the DD specifications for each type of cloning job. Use the line commands that are listed on the panel to access each DD specification panel.
For more information, see [“Verifying DD specifications for the cloning jobs” on page 320](#).
9. When finished, press PF3 (END).
10. If jobs have previously been generated into the specified output data set, a confirmation window is displayed to verify that you want to overwrite the previously generated job. Press PF3 to continue or PF12 to cancel for each job.
11. After the jobs have been generated, the **Member List** panel is displayed.
Jobs and member names are generated based on the type of cloning and the cloning parameters that you set. Therefore, only the jobs and members that are required for the type of cloning that you are performing are displayed. In addition, a LISTDEF member is generated that defines the source objects to be cloned. If data masking is specified as part of the cloning, a MASKDEF member also is generated.
12. Use the Edit or View line commands to review the content of the jobs, if required.
If the contents of the job or member is empty, the job or member will be populated when the source job is submitted.

Verifying DD specifications for the cloning jobs

Db2 Cloning Tool generates the required DDs for each cloning job based on the cloning type and the options that you select for the cloning. You should review each cloning job DD panel to verify the DD specifications.

Before you begin

You must allocate any data sets that do not exist before attempting to run the application cloning jobs.

About this task

Required DD names are pre-selected with SEL in the selection area and have no line command input area. These DDs must be included in the application cloning job. For most DD names, you can modify the data set specifications, such as the DISP or the SYSOUT location.

On any DD specification panel, you can change the individual DD specifications by typing in the input area of the DD. If you want to make changes to the DD data set and member names across all jobs, use the **Default HLQ** and **Default member** fields. Enter the new values in these fields, then enter A in the **Command** field to apply them to all DDs.

Procedure

1. On the **Build Application Clone Jobs** panel, press PF3 (END).
The **DD Specifications for Source Job** panel is displayed.
2. Review the source job DD settings.
3. Enter T in the **Command** field and press Enter.
4. Review the target job DD settings.
5. Review the DD specifications for additional jobs by entering the appropriate line command in the **Command** field.

- There are five DD panels in total, one for each type of cloning job.
6. When finished, press PF3 (END) to build the jobs.

Submitting jobs for application cloning from Db2 source data sets

Once the jobs have been generated, run the jobs that are listed on the **Member List** panel in order.

About this task

You can use the Edit or View line command to review each member that is listed on the panel. Refer to the member description for information about the contents of the member.

If you cannot edit or view a member, it is a placeholder for a member or step that is required. Either the member will be populated after the source job is run, or the member is a placeholder for an external step that you must run separately.

Procedure

1. If the cloning is across LPARs, always run the target TCP/IP server job CKZTG TIP.
2. If the cloning is across LPARs, always run the source TCP/IP server job CKZSRCIP.
3. Run the source job CKZSRC.
4. Ensure that the source job successfully completed.
5. Run the target job CKZTGT.
6. Ensure that the target job successfully completed.

Submitting jobs for application cloning from image copies

Once the jobs have been generated, run the jobs that are listed on the **Member List** panel in order.

About this task

You can use the Edit or View line command to review each member that is listed on the panel. Refer to the member description for information about the contents of the member.

If you cannot edit or view a member, it is a placeholder for a member or step that is required. Either the member will be populated after the source job is run, or the member is a placeholder for an external step that you must run separately.

Procedure

1. If the cloning is across LPARs, always run the target TCP/IP server job CKZTG TIP.
2. If the cloning is across LPARs, always run the source TCP/IP server job CKZSRCIP.
3. Run the source job CKZSRC.
4. Ensure that the source job successfully completed.
5. Run the target job CKZTGT.
6. Ensure that the target job successfully completed.

Submitting jobs for application cloning from user copies

Once the jobs have been generated, run the jobs that are listed on the **Member List** panel in order.

About this task

You can use the Edit or View line command to review each member that is listed on the panel. Refer to the member description for information about the contents of the member.

If you cannot edit or view a member, it is a placeholder for a member or step that is required. Either the member will be populated after the source job is run, or the member is a placeholder for an external step that you must run separately.

Procedure

1. If the cloning is across LPARs, always run the target TCP/IP server job CKZTGIP.
2. If the cloning is across LPARs, always run the source TCP/IP server job CKZSRCIP.
3. Run the source job CKZSRC.
4. Ensure that the source job successfully completed.
5. Set up and run your non-Db2 Cloning Tool job to copy the object data sets from source to target.
Copy the source data sets that contain the data, the SYNCDB2 data set, data sets that contain real-time statistics, and UNLOAD/LOAD data sets (if needed) to the target system. To stop source objects and target objects before the copy, you can include the jobs to stop and start the objects that are generated by Db2 Cloning Tool during the run of the source job. These jobs are listed on the **Member List** panel.
6. When the copy is complete, run the job to start the objects. This job is generated by the source job.
7. Run the job to rename the target objects. This job is generated by the source job.
8. Run the target job CKZTGT.
9. Ensure that the target job successfully completed.

Submitting jobs for application cloning without shared DASD

Once the jobs have been generated, run the jobs that are listed on the **Member List** panel in order.

Before you begin

To use this cloning process, Sterling Connect:Direct must be installed and configured on your system.

About this task

You can use the Edit or View line command to review each member that is listed on the panel. Refer to the member description for information about the contents of the member.

If you cannot edit or view a member, it is a placeholder for a member or step that is required. Either the member will be populated after the source job is run, or the member is a placeholder for an external step that you must run separately.

Procedure

1. Run the target TCP/IP server job CKZTGIP.
2. Run the source TCP/IP server job CKZSRCIP.
3. Run the source job CKZSRC.
4. Ensure that the source job successfully completed.
5. Run the CKZALTD0 job that is generated by the source job to add volume candidates to the target data sets. Running this job prevents potential out-of-space errors that might occur when copying data sets from the source system to the target system.
6. Run the CKZSDCO job to copy Db2 data sets from the source system to the target system using Sterling Connect:Direct. This job is generated by the source job.
7. If you are cloning real-time statistics, run the CKZTCDRO job to copy the RTS data sets from the source to the target system using Sterling Connect:Direct. This job is generated by the source job.
8. Run the target job CKZTGT.
9. Ensure that the target job successfully completed.

Chapter 23. Using the classic ISPF interface

The Db2 Cloning Tool classic ISPF interface allows you to create subsystem and table space cloning jobs using interactive panels. This topic describes how to use the ISPF interface that is available by entering option 9 on the Db2 Cloning Tool Primary Option Menu.

Getting started with the classic ISPF interface

You can use the Db2 Cloning Tool classic ISPF interface to create the JCL and control cards required to clone Db2 subsystems and to clone table spaces and index spaces. The menu-driven interface allows you to easily create cloning jobs with specific command parameters, and then save that information in profiles that can be used again. In addition, subsystem information can be configured once and then is available to all users of the interface.

Starting the interface

Note: The Db2 Cloning Tool classic ISPF interface requires a minimum region size of 50000 KB.

Note: Before attempting to use the classic ISPF interface, ensure that customization steps been completed as described in [Chapter 4, “Customizing Db2 Cloning Tool,”](#) on page 77 and [Chapter 5, “After customizing Db2 Cloning Tool using Tools Customizer,”](#) on page 97.

Start the interface by entering option 9 on the Db2 Cloning Tool Primary Option Menu.

The Db2 Cloning Tool Primary Option Menu

The Db2 Cloning Tool **Primary Option Menu** is the starting point for all cloning functions using the classic ISPF interface. The **Primary Option Menu**, shown in the following figure, is displayed when you start the Db2 Cloning Tool classic ISPF interface.

```
Db2 Cloning Tool for z/OS   Primary Option Menu
Option ==>

0  User settings           User ID . . : TWUSR
1  Clone                  System ID . : LPAR
2  Administrator functions Appl ID . . : CKZ
X  Exit                   Version . . : 3.2
```

From the **Primary Option Menu**, you can perform the following actions by entering the corresponding option number in the **Option** field and pressing Enter:

0 - User settings

Select this option to specify defaults for command parameters, work data sets, other settings for both subsystem cloning and table space cloning. These defaults are used for creating cloning profiles.

1 - Clone

Select option 1 to create cloning profiles for subsystem or table space cloning.

2 - Administrator functions

Select option 2 to add or configure Db2 subsystems to be used as source and targets for the cloning process. Db2 subsystems should be configured before attempting to create cloning profiles.

X - Exit

Select this option to exit the ISPF interface.

About cloning profiles

All of the settings that are required to build the jobs for subsystem or table space cloning are saved in VSAM profile repository. You can create profiles that can be shared with other users, or you can specify that profiles be read-only or completely inaccessible by others.

Note: Application cloning profiles cannot be used in the classic ISPF interface.

Note: If you open a cloning profile that was created in Db2 Cloning Tool V3.1 using the Db2 Cloning Tool V3.2 ISPF interface, the profile is converted to V3.2 and can no longer be used in V3.1. For information about porting V3.1 profiles to V3.2, see the ISPF interface tasks section in the topic [“Worksheets: Gathering parameter values for Tools Customizer”](#) on page 31.

About the ISPF help system

ISPF help panels are available. Every product panel in Db2 Cloning Tool has its own help panel. The help panel lists the purpose of the panel, available commands, and fields and column data that is displayed. Enter HELP or press PF1 to display a help panel.

Detailed information about command parameters and other valid values on the panels are provided in the help panels. The panel values are not described in detail in these topics. Use the help system or the command reference topics that follow if you need more information.

- [Chapter 26, “Db2 Cloning Tool Subsystem Cloning commands,”](#) on page 371
- [Chapter 27, “Db2 Cloning Tool Table Space Cloning commands,”](#) on page 513

Configuring Db2 subsystems

The Administrator functions option on the **Primary Option Menu** allows you to configure all Db2 subsystems that might be used by subsystem or table space cloning procedures.

You must define the Db2 subsystem information for your site on these panels before attempting to create subsystem or table space cloning jobs.

Enter 2 on the Db2 Cloning Tool **Primary Option Menu** menu to access administrator functions. The **Administrator functions** panel is displayed, as show in the following figure:

```
Db2 Cloning Tool for z/OS      Administrator functions
Option ==>
1  DB2 subsystems
```

Configuring a subsystem

To create a new subsystem or configure a subsystem that you have already created, follow these steps.

Procedure

1. On the **Administrator functions** menu, enter option 1.
2. On the **DB2 subsystems** panel, enter C in the Command field.
The **Enter New DB2 Subsystem Profile Options** window is displayed.
3. On the **Enter New DB2 Subsystem Profile Options** window, enter the Db2 subsystem ID.
4. Press Enter.
The **Edit DB2 Subsystem** panel is displayed.
5. Specify Db2 load library names for the subsystem being defined.
6. If the subsystem will be used for subsystem cloning, enter 1 in the Option line to specify information required for cloning Db2 subsystems and press Enter.
7. If the subsystem will be used for table space cloning, enter 2 in the Option line to specify information required for table space cloning and press Enter.

Specifying information for subsystem cloning

To successfully generate JCL that uses this subsystem as a target for subsystem cloning jobs, you must provide the system VCAT and the special ZPARMs member.

Refer to the product help panel for more detailed information about the fields on this panel.

Enter **1** on the **Edit DB2 Subsystem** menu. The **Subsystem cloning information** panel is displayed, as shown in the following figure:

```
Subsystem cloning information
Command ==>

Commands: S - Static aliases  D - Dynamic aliases

SSID . . . . . : SS01
Description . . . . . :

Use as Subsystem Cloning Source or Target only . . . . . More: +
                                                         (SOURCE, TARGET,
                                                         or blank)

System ID where this DB2 normally runs . . . . .

Group name . . . . . (if data sharing)
Member name . . . . . (if data sharing)
Special ZPARMs member . . SS01ZPRM
System VCAT . . . . . SS01VCAT

DDF:
LOCATION . . . . .
GENERIC . . . . .
LUNAME . . . . .
PASSWORD . . . . .
PORT . . . . . (1-65535, or blank)
RESPORT . . . . . (1-65535, or blank)
SECPORT . . . . . (1-65535, or blank)
IPNAME . . . . .
IPV4 . . . . .
IPV6 . . . . .
GRPIP4 . . . . .
GRPIP6 . . . . .
```

Specifying information for table space cloning

This topic describes how to specify subsystem information for Db2 table space cloning in the administrative options.

Enter **2** on the **Edit DB2 Subsystem** menu. The **Tablespace cloning information** panel is displayed:

```
Tablespace cloning information
Option ==>

SSID . . . . . : SS01
Description . . . . . :

System ID where this DB2 normally runs . . . . .

Group name . . . . . (if data sharing)
Group attach name . . . . . (if data sharing)
Member name . . . . . (if data sharing)
Member ID . . . . . (if data sharing, 1-32, or blank)
Default VCAT . . . . .

TCPIP Server:
PORT . . . . . (1-65535, or blank)
IPV4 . . . . .
IPV6 . . . . .
```

Refer to the product help panel for more detailed information about the fields on this panel.

Configuring user settings

The User Settings option on the **Primary Option Menu** allows you to specify defaults that are used when creating profiles. Defaults can be set for commands, work data sets, and job cards for both subsystem cloning and table space cloning.

Defaults are originally derived from the CKZINI PARMLIB member, but can be customized for each user ID using these panels. The defaults are saved in a VSAM profile repository and are specific to each TSO user ID and LPAR. This allows different users to have different defaults when creating cloning profiles.

Enter 0 on the Db2 Cloning Tool **Primary Option Menu** to access user options. The **User Settings** panel is displayed, as shown in the following figure:

```
Db2 Cloning Tool for z/OS      User Settings
Option ==>

0  User Options
1  User DB2 subsystem clone settings
2  User DB2 tablespace clone settings
```

Setting job card defaults

Follow these steps to set job card defaults that apply to both volume and table space cloning.

Procedure

1. Enter 0 on the **User Settings** menu to access user options.
The **Set Processing Options** menu is displayed.
2. On the **Set Processing Options panel**, enter **1** to specify job card options and press Enter.
The **Set Batch Job Card Information** panel is displayed.
3. Enter job card information for your site.
4. Press **PF3 (END)** to return to the **Set Processing Options** panel.

Setting subsystem cloning defaults

This topic describes how to specify defaults for subsystem cloning commands and work data sets.

About this task

Defaults are originally derived from the SCKZPARM (CKZINI) member, but can be customized for each user ID using these panels. The defaults are saved in a VSAM profile repository and are specific to each TSO user ID and LPAR. This allows different users to have different defaults when creating cloning profiles.

Procedure

1. Enter **1** on the **User Settings** menu to access subsystem cloning defaults. The **User DB2 subsystem clone settings** panel is displayed, as shown in the following figure:

```

User DB2 subsystem clone settings
Command ==>>>

Commands: R - Reset

Subsystem Clone Profile Default Values:
Prefix for work data sets . . DHWRK
Work data sets unit device . . (SYSDA, DISK, or etc.)

Valid command selection values are
1 COPY command
2 COPYCHECK command
3 RENAME command
4 DB2FIX command
5 DB2SQL command
6 DB2START command
7 DB2STOP command
8 DB2UPDATE command
9 DB2UTILXCLEAN command
10 RESTORE-FROM-DUMPTAPES command

```

2. Set work data set values as required.
3. To set default values for each command, enter the appropriate number for the command in the **Command** line. Refer to the help system for detailed information about the fields for subsystem cloning defaults.

What to do next

To reset all default values for subsystem cloning commands to installation defaults, which are supplied in the SCKZPARM (CKZINI) member:

1. Enter R in the **Command** field and press Enter.
2. A window prompts you to confirm the reset of all defaults to installation-supplied values. Press Enter or PF3 to continue, or press PF12 to cancel.

Setting table space cloning defaults

This topic describes how to specify defaults for table space cloning commands, DDs, and other defaults.

About this task

Defaults are originally derived from the CKZINI PARMLIB member, but can be customized for each user ID using these panels. The defaults are saved in a VSAM profile repository and are specific to each TSO user ID and LPAR. This allows different users to have different defaults when creating cloning profiles.

Procedure

1. Enter **2** on the **User Settings** menu to access table space cloning defaults. The **User tablespace clone settings** panel is displayed:

```

User DB2 tablespace clone settings
Command ==>>>

Commands: R - Reset

1 DD Specification
2 SET Command
3 COPY Command
4 HLQDDDF Command

```

- To set default values for each command, enter the appropriate number for the command in the **Command** line. Refer to the help system for detailed information about the fields for table space cloning defaults.

What to do next

To reset all default values for table space cloning commands to installation defaults, which are supplied in the SCKZPARM (CKZINI) member:

- Enter R in the **Command** field and press Enter.
- A window prompts you to confirm the reset of all defaults to installation-supplied values. Press Enter or PF3 to continue, or press PF12 to cancel.

Setting default DD specifications for table space cloning

The **DB2 tablespace clone DD Specification** panel lets you enter default DD specifications for the DDs required for table space cloning and for optional user DDs. These DDs may be used in one or more of the source, target, or TCPIP jobs.

Note: You must allocate any data sets that do not exist before attempting to execute the table space cloning jobs.

DD defaults can be customized for each user ID using these panels. The defaults are saved in a VSAM profile repository and are specific to each TSO user ID and LPAR. This allows different users to have different defaults when creating cloning profiles.

Enter **1** on the **User DB2 tablespace clone settings** menu. The **DB2 tablespace clone DD Specification** panel is displayed:

```

DB2 tablespace clone DD Specification
Command ===>                               Scroll ===> PAGE

Commands:      D - Set Defaults  C - Clear Defaults  U - User DD Specification
Line commands: S - Select/Unselect

Control DD defaults:
HLQ . . . . TWUSR                          (control HLQ)
Member . . . LSTDMBR                       (control member)

Row 1 of 19

  DD Name  DD
SEL CKZIN  *
SEL CKZPRINT  SYSOUT=*
SEL CKZINI  DISP=SHR,DSN=CKZ.WRK0310.SCKZPARM(CKZINI)
SEL CKZLOG  SYSOUT=*
SEL CKZLSTDF  DISP=SHR,DSN=TWUSR.LISTDEF(LSTDMBR)
  CKZMSKDF  DISP=SHR,DSN=TWUSR.MASKDEF(LSTDMBR)
  CKZCRXML  DISP=SHR,DSN=TWUSR.XMLCRDDL(LSTDMBR)
SEL CKZSYNC  DISP=OLD,DSN=TWUSR.SYNCCB2(LSTDMBR)
  CKZCOPY  DISP=OLD,DSN=TWUSR.COPYDSNS(LSTDMBR)
  CKZZXML  DISP=OLD,DSN=TWUSR.XMLSTR(LSTDMBR)
  CKZSQL  DISP=OLD,DSN=TWUSR.SQLOUT(LSTDMBR)
  CKZSTPT  DISP=OLD,DSN=TWUSR.CMDSSTPT(LSTDMBR)
  CKZSTPS  DISP=OLD,DSN=TWUSR.CMDSSTPS(LSTDMBR)
  CKZSTRS  DISP=OLD,DSN=TWUSR.CMDSSTRS(LSTDMBR)
  CKZIDCAM  DISP=OLD,DSN=TWUSR.IDCAMS(LSTDMBR)
  CKZDDL  DISP=OLD,DSN=TWUSR.DDLLOUT(LSTDMBR)
  CKZRRJOB  DISP=OLD,DSN=TWUSR.RRJOB
  CKZRRDSN  DISP=OLD,DSN=TWUSR.RRDSN
SEL CKZERROR  SYSOUT=*
  SYSINCKZ  DISP=SHR,DSN=TWUSR.LOGAPCTL(LSTDMBR)
  SYSOUT  SYSOUT=*
SEL SYSUDUMP  SYSOUT=*

```

Required DD names are pre-selected with SEL next to of the DD name and will be included in one or more of the Db2 Cloning Tool Table Space Cloning jobs (source, target, TCP/IP server, or source TCP/IP server).

For most DD names, you can modify the data set specifications such as DISP or the SYSOUT location. Some DD names are not required, but when specified must have a particular DD name. These DD names cannot be modified.

Refer to the information that follows to determine which DDs to include in your table space cloning jobs.

Editing DD names and specifications

Where allowed, you can change the DD names and specifications to meet your site's requirements. You can type directly over the current values in the fields.

Using the Control DD defaults fields to edit DD specifications

The **Control DD** fields can be used to easily set the high level qualifiers and member names for all the DDs on the panel that can be modified.

1. Specify a default high level qualifier in the **HLQ** field and a default member name in the **Member** field.
2. Enter D in the **Command** field
3. When you press Enter, the DD name fields are populated with the specified HLQ and member name.

To clear the DD specifications, enter C in the **Command** field.



Attention: if you use the C command, all specifications that will be modified will be cleared.

Selecting or deselecting a DD for inclusion in JCL

Use the S line command as a toggle to select or deselect a DD. If the DD will be included, SEL is displayed next to the DD name. If a DD is not selected, it will not be included by default in the table space cloning profile, but can be added later when creating a table space cloning profile.

Adding user DDs

You can enter your own user-defined DD names and specifications on a separate panel. Enter the U command in the **Command** field. When you press Enter, the following panel is displayed:

```
Command ==>          DB2 tablespace clone DD Specification          Scroll ==> PAGE
Commands:      A - Add Line  P - Product DD Specification
Line commands: S - Select/Unselect  D - Delete Line
               DD Name  DD
```

This panel allows you to enter default DD specifications for user DDs that you may want to include in table space cloning jobs.

To add DDs to this panel:

1. Enter A in the Command line and press Enter.
2. In the lines that are displayed in the input area, enter the DD name and desired data set specifications.
3. Enter S next to the DD name to select the DD for inclusion in jobs.

Use the S line command as a toggle to select or deselect a DD for inclusion. If a DD is not selected, it will not be included by default in the table space cloning profile, but can be added later when creating a table space cloning profile.

To remove a DD, enter D next to the DD and press Enter.

To return to specifying table space cloning DDs, enter P in the command line.

Table space cloning DD descriptions

The table space cloning DDs provided by Db2 Cloning Tool Table Space Cloning on the **DB2 tablespace clone DD Specification** panel are identified in this topic.

The following table describes the table space cloning DDs.

| <i>Table 66. Table space cloning DD descriptions on the DB2 tablespace clone DD Specification panel</i> | | | |
|--|-----------|--|---|
| DD | Required? | Usage | Default |
| CKZIN | Yes | In the source job, it identifies source (local) and target Db2 subsystem names, identifies DDs passed to ADRDSSU for data set allocations, and contains other commands related to the source and target subsystems. The CKZIN DD also contains the COPY command and its options, and various SET commands. | * |
| | | In the target job, the CKZIN DD contains the input parameters for the target job. It must point to the data set referenced by the CKZSYNC DD in the source job DD specification. | |
| | | In the TCP/IP server job, the CKZIN DD contains the input parameters for the TCP/IP job in the form of SET commands. | |
| CKZPRINT | Yes | In the source job, CKZPRINT displays CKZINI tokens, control parameters, data set names and associated Db2 table spaces and index spaces, Db2 start and stop space command status, and DFSMSdss program ADRDSSU commands and status. | SYSOUT=* |
| | | In the target job, CKZPRINT displays CKZINI tokens, CKZIN control parameters, Db2 SQL execution status, and SYNCDB2 status and START Db2 command status for each data set processed. | |
| CKZINI | Yes | CKZINI is the product PARMLIB member that is set up during installation. Used in the source and the target jobs, the CKZINI member contains program variables. | The PARMLIB library where the PARMLIB member CKZINI is located. |
| CKZLOG | No | In the source job, CKZLOG displays LISTDEF processing, Db2 commands issued, and the responses or results of the commands. | SYSOUT=* |
| | | In the target job, CKZLOG displays the Db2 commands issued and the responses or results of the commands. It also displays detailed information about each Db2 page access. | |

Table 66. Table space cloning DD descriptions on the **DB2 tablespace clone DD Specification** panel (continued)

| DD | Required? | Usage | Default |
|----------|---|---|--|
| CKZLSTDF | Yes | In the source job, CKZLSTDF contains LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be processed and, if requested, copied to the target. | DISP=SHR, DSN=hlq-field. LISTDEF(member-field) |
| CKZMSKDF | Yes, if data masking will be used | CKZMSKDF is used in the source job to hold the masking rules to be applied and the tables to be masked. It is passed to the target job as an input to the data masking processor. | DISP=SHR, DSN=hlq-field. MASKDEF(member-field) |
| CKZCRXML | Yes if cloning tables that contain XML data | CKZCRXML is not used in the source job. In the target job, this DD contains the DDL to create an XML table for adding strings to SYSIBM.SYSXMLSTRINGS using an SQL INSERT. The data set pointed to by the CKZCRXML DD must have an LRECL of 80 and RECFM of FB. | DISP=SHR, DSN=hlq-field. XMLCRDDL(member-field) |
| CKZSYNC | Yes in most cases | In the source job, CKZSYNC is an output data set that will be used by the target job to make the VSAM objects accessible on the target Db2 subsystem. | DISP=OLD, DSN=hlq-field. SYNCDB2(member-field) |
| CKZCOPY | No | In the source job, CKZCOPY points to a data set which will contain a list of TO and FROM data set names derived from the LISTDEF command input. This DD is provided to assist in copying the selected VSAM objects outside of Db2 Cloning Tool Table Space Cloning and is only used if the keyword for the COPY command is DATA-MOVER(PGM(NONE)). | DISP=OLD, DSN=hlq-field. COPYDSNS(member-field) |
| CKZXML | No; this DD is obsolete and cannot be selected. | In the source and target jobs, CKZXML is used to pass the source subsystem XMLSTRINGS catalog table contents from the source job to the target job. | DISP=OLD, DSN=hlq-field. XMLSTR(member-field) |
| CKZSQL | No; this DD is obsolete and cannot be selected. | In the source and target jobs, CKZSQL is used to pass SQL commands to the target job to synchronize identity column values between the target and source. | DISP=OLD, DSN=hlq-field. SQLOUT(member-field) |
| CKZSTPT | No | In the source job, CKZSTPT is used to submit Db2 commands to stop the target Db2 table spaces and index spaces and to deallocate the target VSAM data sets. | DISP=OLD, DSN=hlq-field. CMSSTPT(member-field) |
| CKZSTPS | No | In the source job, CKZSTPS is used to submit Db2 commands to stop the source Db2 table spaces and index spaces and to deallocate the source VSAM data sets. | DISP=OLD, DSN=hlq-field. CMSSTPS(member-field) |

Table 66. Table space cloning DD descriptions on the **DB2 tablespace clone DD Specification** panel (continued)

| DD | Required? | Usage | Default |
|----------|--|---|--|
| CKZSTRS | No | In the source job, CKZSTRS is used to submit Db2 commands to start the source Db2 table spaces and index spaces after they have been copied to the target. | DISP=OLD, DSN=hlq-field. CMDSSTRS(member-field) |
| CKZIDCAM | No | In the source job, CKZIDCAM is used to output IDCAMS deletes and renames for all data sets that were copied as .F0001 data sets, rather than .I0001 or .J0001. This might occur if some or all target table spaces or index spaces did not exist in the Db2 catalog before the source job is run. This job can then be used as input to the CKZFIX job to correct missing target data sets. | DISP=OLD, DSN=hlq-field. IDCAMS(member-field) |
| CKZDDL | Yes | The CKZDDL is used in the source job. Depending on the type of DDL processing specified, it contains source or target object DDL. It also can contain ALTER TABLE SQL for identity columns. | DISP=OLD, DSN=hlq-field. DDLOUT(member-field) |
| CKZRRJOB | No | If using the target job runtime repository for rerunning the target job or keeping a job history, this DD will be included in the target and report jobs. | DISP=OLD, DSN=hlq-field. RRJOB |
| CKZRRDSN | No | If using the target job runtime repository for rerunning the target job or keeping a job history, this DD will be included in the target and report jobs. | DISP=OLD, DSN=hlq-field. RRDSN |
| CKZERROR | No | When CKZERROR is included in source, target, and/or TCP/IP server jobs, all warning and error messages are output to this DD, as well as to CKZPRINT. | SYSOUT=* |
| SYSINCKZ | Yes if image copies will be used as the source for the cloning | This DD is used for log apply processing when image copies are used as the source for the table space cloning. | DISP=SHR, DSN=hlq-field. LOGAPCTL(LSTDMBR) |
| SYSOUT | Yes if image copies will be used as the source for the cloning | This DD is used for message output when image copies are used as the source for the table space cloning. | SYSOUT=* |

Table 66. Table space cloning DD descriptions on the **DB2 tablespace clone DD Specification** panel (continued)

| DD | Required? | Usage | Default |
|----------|--|--|---|
| CKZULREC | Yes, if the UNLOAD-LOAD on mismatch function will be used for the cloning | <p>If you specify UNLOAD-LOAD-ENABLE(Y) for table spaces that are ineligible for copy processing, this DD will be included in the source job for the SYSREC TEMPLATE control statement.</p> <p> Attention: You must use the CKZULREC member from the SCKZJCL sample library for this DD. Before cloning, copy the member to the data set that is specified for the CKZULREC DD (such as DSN=h1q-field.ULREC(member-field)). You can modify the sample member if required.</p> | DISP=SHR, DSN=h1q-field.ULREC(member-field) |
| CKZULPUN | Yes, if the UNLOAD-LOAD on mismatch function will be used for the cloning. | <p>If you specify UNLOAD-LOAD-ENABLE(Y) for table spaces that are ineligible for copy processing, this DD will be included in the source job for the SYSPUNCH TEMPLATE control statement.</p> <p> Attention: You must use the CKZULPUN member from the SCKZJCL sample library for this DD. Before cloning, copy the member to the data set that is specified for the CKZULPUN DD (such as DSN=h1q-field.ULPUN(member-field)). You can modify the sample member if required.</p> | DISP=SHR, DSN=h1q-field.ULPUN(member-field) |
| CKZULOUT | Yes, if the UNLOAD-LOAD on mismatch function will be used for the cloning. | <p>If you specify UNLOAD-LOAD-ENABLE(Y) for table spaces that are ineligible for copy processing, this DD will be included in the source job for the SORTOUT TEMPLATE control statement.</p> <p> Attention: You must use the CKZULOUT member from the SCKZJCL sample library for this DD. Before cloning, copy the member to the data set that is specified for the CKZULOUT DD (such as DSN=h1q-field.ULOUT(member-field)). You can modify the sample member if required.</p> | DISP=SHR, DSN=h1q-field.ULOUT(member-field) |

Table 66. Table space cloning DD descriptions on the **DB2 tablespace clone DD Specification** panel (continued)

| DD | Required? | Usage | Default |
|----------|--|---|---|
| CKZULUT | Yes, if the UNLOAD-LOAD on mismatch function will be used for the cloning. | <p>If you specify UNLOAD-LOAD-ENABLE(Y) for table spaces that are ineligible for copy processing, this DD will be included in the source job for the SYSUT TEMPLATE control statement.</p> <p> Attention: You must use the CKZULUT member from the SCKZJCL sample library for this DD. Before cloning, copy the member to the data set that is specified for the CKZULUT DD (such as DSN=h1q-field.ULUT(member-field)). You can modify the sample member if required.</p> | DISP=SHR,DSN=h1q-field.ULUT(member-field) |
| SYSUDUMP | No | <p>The SYSUDUMP DD is used for diagnostic purposes in all cloning jobs for the first data capture when an abend occurs.</p> <p>The SYSUDUMP DD is used only when abends occur. At a minimum, it is recommended to specify the following parameters for SYSUDUMP in PARMLIB:</p> <pre>OPTIONS SDATA=(SUM), PDATA=(SA,REGS,LPA,JPA,PSW,SPLS,SUBTASKS)</pre> <p>All of these parameters should be specified to ensure that there is enough data for the first data capture to facilitate diagnosis when an abend occurs. If you would like to control the size of the job output, then you can use the JOB card or the JES2 /*JOBPARM statement or the JES3 // *MAIN statement parameters to limit job output.</p> | SYSOUT=* |

Creating cloning jobs using the interface

This topic describes some basic procedures for subsystem cloning and table space cloning using the ISPF interface.

Before you begin

Before you begin creating subsystem or table space cloning jobs using the ISPF interface:

- You must ensure that the subsystems that you want to use as source and target subsystems have been added and configured using the Administrator functions portion of the interface.
- You should verify that the user configuration settings are appropriate for your needs.

Basic procedure

The basic procedure to create cloning jobs for both subsystem and table space cloning is:

1. Create a cloning profile.
2. Select the menu options to specify the type of cloning, the DDs, command settings, and other options.
3. Build the profile. Building the profile creates the series of jobs that will perform the cloning.
4. Review the generated jobs to ensure the output is as desired.
5. To invoke the cloning process, submit the generated jobs. The member names are generated alphanumerically; simply submit the jobs in order.

Creating a profile

To create a new subsystem or table space cloning profile, follow these steps.

About this task

The process of creating a profile is the same for subsystem cloning and table space cloning.

Procedure

1. On the **Primary Option** menu, enter option 1.
2. On the **Clone** menu, enter option 1 for subsystem cloning or 2 for table space cloning.
3. On the **Enter Clone Profile Selection Criteria** window, enter selection criteria (if desired). Standard ISPF wildcarding is allowed.
4. On the **DB2 Subsystem Clone Profile Display** or the **DB2 Tablespace Clone Profile Display**, enter C in the Command field.
The **Enter New DB2 Subsystem Clone Profile Options** or **Enter New DB2 Tablespace Clone Profile Options** window is displayed.
5. Enter a profile name and share option and press Enter.
The **Edit DB2 Subsystem Clone Profile** menu or the **Edit DB2 Tablespace Clone Profile** menu is displayed.
6. To set up the cloning profile, select the options on the menu.

Results

Once a profile has been created, it can be edited, renamed, viewed, copied or deleted.

Subsystem cloning

This section describes the basic steps for subsystem cloning using the ISPF interface.

Start the ISPF interface and create a subsystem cloning profile. After the cloning profile has been created, the **Edit DB2 Subsystem Clone Profile** menu is displayed, as shown in the following figure:

```

                                Edit DB2 Subsystem Clone Profile
Option ==>>

Creator . . . : TWUSR           Name . . . . : TEST2
Share Option . : UPDATE        Description . .

1 Select Source and Target DB2 subsystems
2 Select Source and Target Volume Pairing
3 Select Source and Target ICF catalogs
4 Select Rename masks
5 Select other parameters

```

Subsystem cloning steps summary

The general steps for using the ISPF interface to clone a Db2 subsystem are described in this topic.

Procedure

1. Ensure the source and target subsystems have been added and properly configured under the Administrative Options option.
2. Create a subsystem cloning profile.
3. Select the source and target subsystems and specify the cloning type (option 1 on the **Edit DB2 Subsystem Clone Profile** panel).
4. Verify or enter source and target HLQs (option 1 on the **Edit DB2 Cloning Profiles** panel).
5. Specify source and target stogroup or volume pairings (option 2 on the **Edit DB2 Subsystem Clone Profile** panel).
6. Specify source and target ICF catalogs (option 3 on the **Edit DB2 Subsystem Clone Profile** panel).
7. Specify rename masks (option 4 on the **Edit DB2 Subsystem Clone Profile** panel).
8. Build the profile.
9. Submit the jobs (in order).

Step summary for subsystem cloning from a system-level backup

The general steps for using the ISPF interface to clone a Db2 subsystem from a system level backup are described in this topic.

Procedure

1. Ensure the source and target subsystems have been added and properly configured under the Administrative Options option.
2. Create a subsystem cloning profile.
3. On the **Edit DB2 Subsystem Clone Profile** panel, enter option 1 to specify the source and target subsystems and specify cloning type of online.
4. On the **Edit DB2 Subsystem Clone Profile** panel, enter option 2 to specify source and target volume pairing.
5. On the **Select Source and Target Volume Pairing** panel, enter option 3 to select Source System Level Backup.
6. On the **Set Source System Level Backup** panel, enter DB2SLB in the **System Level Backup type** field.
7. To specify source and target volume pairing, on the **Set Source System Level Backup** panel, do one of the following:
 - To use storage group names, enter YES in the **Pair using Source Storage Group names** field. You must also specify the source and target storage groups by using options 1 and 4 on the **Select Source and Target Volume Pairing** panel.
 - To use volume serials, enter NO in the in the **Pair using Source Storage Group names** field. You must also specify either target volumes or target storage groups by using options 4 or 5 on the **Select Source and Target Volume Pairing** panel.
8. If you want to use SLB dump tapes as the source for the cloning:
 - a) Enter YES in the **Use Dumptapes** field.
 - b) Enter R in the Command field.
 - c) On the **RESTORE-FROM-DUMPTAPES** command panel, verify the settings for the RESTORE-FROM-DUMPTAPES command.
9. Build the profile.
10. Submit the jobs (in order).

Select source and target subsystems

You must first select the source and target subsystems and specify offline or online cloning.

For non-data sharing source subsystems

Select the source and target subsystems and specify offline or online cloning.

Procedure

1. On the **Edit DB2 Subsystem Clone Profile** menu, enter option 1.
2. On the **Select Source and Target DB2 Subsystems** menu, enter A in the Command field.
3. On the **Select Source DB2 Subsystem** panel, select the source subsystem and press Enter.
4. On the **Select Cloning Type** panel, enter ONLINE or OFFLINE in the **Type of cloning** field.
5. Press Enter.
The **Select Target DB2 Subsystem** panel is displayed.
6. Select a target subsystem and press Enter.
The **Edit DB2 cloning values** panel is displayed.

For data sharing source subsystems

Select the source and target subsystems and specify offline or online cloning. In addition, for source subsystems that are data sharing, you must specify the data sharing attributes of the target subsystem.

Procedure

1. On the **Edit DB2 Subsystem Clone Profile** menu, enter option 1.
2. On the **Select Source and Target DB2 Subsystems** menu, enter A in the Command field.
3. On the **Select Source DB2 Subsystem** panel, select the source subsystem and press Enter.
4. On the **Select Cloning Type** panel, enter ONLINE or OFFLINE in the **Type of cloning** field.
5. Enter SAME, FEWER, or NONDS in the **Data sharing attributes of target** field.
Specify SAME when the target will be a data sharing group with the same number of members as the source. Specify FEWER if the target will be a data sharing group with fewer members than the source. Enter NONDS when the target will not be a data sharing group.
6. The **Select source members to clone** panel is displayed. This panel lists all data sharing group members in the source data sharing group. The data sharing group member that you selected as a source subsystem on the **Select Source DB2 Subsystem** panel is automatically selected. You can select one or more additional data sharing group members to clone.
7. When you have selected all the members to be cloned, press Enter.

Specifying the target subsystem for data sharing source members

You must specify which members of the target data sharing group are to be used as targets.

If the target data sharing group will have the SAME number of members

1. **The Select Target DB2 member** panel is displayed. On this panel, select the target subsystems to be paired with source subsystems. The target subsystem you select will be paired with the source member listed near the top of the panel.
2. Select a target by entering S next to the target. When you press Enter, if there are more source subsystems that need to be paired with targets, the panel will be displayed again.
3. Continue to pair target subsystem with source subsystems until all source and targets have been paired; when you press Enter, the **Edit DB2 Cloning values** panel is displayed.

If the target data sharing group will have FEWER members

If the target is a data sharing group that will have fewer members than when built, then you must specify which target members will be the “surviving members”. For example, if you plan to clone one subsystem to a target data sharing group that has two members, then you will need to specify which target group member will be the surviving member.

1. The **Select Target DB2 member** panel is displayed. On this panel, select the target subsystems to be paired with source subsystems. The target subsystem you select will be paired with the source member listed near the top of the panel.
2. Select a target by entering S next to the target. When you press Enter, if there are more source subsystems that need to be paired with targets, the panel will be displayed again. Continue to pair target subsystem with source subsystems until all source and targets have been paired.
3. The **Manage surviving target DB2 members** panel is displayed. On this panel, you can add or delete the surviving member(s) for the target data sharing subsystem. Add a surviving target member by entering A in the Command line and press Enter.
4. The **Select surviving target DB2 members** panel is displayed. Select the surviving member by entering S in the line command area next to the target subsystem. Or select all listed members by entering A in the command line. Press Enter.
5. The **Manage surviving target DB2 members** panel is displayed. Each selected surviving member is listed on the panel. The **PRIMARY** and **DDF** fields default to SOURCE, which means the target subsystems retain the BSDS attributes of the subsystems. If required, you can edit those fields on this panel.
6. To save changes, press Enter, then PF3. The **Enter DB2 cloning values** panel is displayed.

If the target subsystem will not be a data sharing group (NONDS)

1. The **Select Target DB2 member** panel is displayed. On this panel, select the target subsystems to be paired with source subsystems. The target subsystem you select will be paired with the source member listed near the top of the panel.
2. Select a target by entering S next to the target. When you press Enter, if there are more source subsystems that need to be paired with targets, the panel will be displayed again. Continue to pair target subsystem with source subsystems until all source and targets have been paired.
3. When you press Enter, the **Select surviving target DB2 member panel** is displayed. This panel lists the target subsystem you selected. Since the target will be non-data sharing, this is the only subsystem that will be a surviving member. Enter S in the line command area to select the surviving member.
4. When you press Enter, the **Edit DB2 Cloning values** panel is displayed.

Add or verify high level qualifiers

The high level qualifiers of the Db2 source and target subsystems must be provided. Verify that they have already been defined when the subsystems were configured under administrative options, or provide the qualifiers using the **Enter DB2 HLQs** panel.

Procedure

1. On the **Edit DB2 cloning values** menu, enter option 1.
2. On the **Enter DB2 HLQs** panel, verify or enter the Db2 HLQs for the source and target subsystem(s).

Specify source and target volume pairings

Specify the input volumes to be copied and the target volumes to which they will be copied.

Procedure

1. On the **Edit DB2 Subsystem Clone Profile** menu, enter option 2.
2. On the **Select Source and Target Volume Pairing** panel, enter one of the following:
 - 1 to specify a source SMS storage group or mask that contains the input volumes to be copied.
 - 2 to specify input volumes using volsers or volser masks.
 - 3 to specify a system level backup (SLB) as input for the source
 - 4 to specify one or more SMS storage groups or masks as targets that will be paired with input volumes.
 - 5 to specify the target volumes via volsers or volser masks that will be paired with input volumes.
 - 6 to specify source volumes that are to be excluded from cloning.
 - 7 to specify target volumes that are to be excluded from cloning.

Specify source and target ICF catalogs

Specify the source catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalogs that renamed volume data sets are to be cataloged in.

About this task

For each renamed data set, source and target catalog pairs are searched for the ICF catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

Procedure

1. On the **Edit DB2 Subsystem Clone Profile** menu, enter option 3.
2. On the **Select Source and Target ICF catalogs** panel, specify the source catalogs that data sets from the source volume are cataloged in, and the corresponding target catalog that renamed data sets on the target volume are to be cataloged in.

Specify rename masks for source and target data sets

The data sets from the COPY step can be renamed onto the target volumes.

About this task

For each renamed data set, source and target catalog pairs are searched for the ICF catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

Procedure

1. On the **Edit DB2 Subsystem Clone Profile** menu, enter option 4 (Select rename masks).
2. On the **Select rename masks** panel, enter one of the following:
 - 1 to specify source and target rename masks. On the **Rename Masks** panel, specify the source and target data set rename masks. The RENAME uses these masks to rename and catalog the data sets from the COPY step onto the target volumes.
 - 2 to exclude data sets from the rename process. The **Exclude Masks** panel allows you to specify a list of source data set names or masks that will NOT be renamed.

Build the cloning jobs from a profile

Once the profile has been created, build the profile to produce the cloning jobs.

About this task

Begin building the profile on the **DB2 Subsystem Clone Profile Display**, shown in the following figure:

```
DB2 Subsystem Clone Profile Display                               Scroll ==> CSR
Command ==>
Commands:      C - Create
Line Commands: B - Build D - Delete E - Edit R - Rename V - View C - Copy

Profile Like . . . *
Creator Like . . . TWUSR*

Row 1 of 2
Cmd Name      Creator  Share Option  Description
TEST         TWUSR   NO ACCESS
TEST DATA SHARING TWUSRA  UPDATE       TEST DATA SHARING #1
```

Procedure

1. On the **DB2 Subsystem Clone Profile Display**, enter B next to the profile that you want to build.
2. On the **Build DB2 Subsystem Clone jobs** panel, enter the data set into which the jobs are to be placed, and specify processing options.
3. Press Enter.

The cloning jobs are generated into the specified data set.

4. If you selected the processing option to edit the JCL data set, a panel is displayed listing the jobs that have been generated, as shown in the following figure. You can edit or view the jobs using line commands.

```
Edit TWUSR.CKZ.JCLLIB4                                         Scroll ==> CSR
Command ==>
Line Commands: E - Edit V - View

Row 1 of 22
Cmd Name      Created      Changed      ID
ST01C7A1     2009/08/25  2009/08/25  08:59:50  TWUSR
ST01D8A     2009/08/25  2009/08/25  08:59:53  TWUSR
ST02        2009/08/25  2009/08/25  08:59:52  TWUSR
ST03C7A1     2009/08/25  2009/08/25  08:59:50  TWUSR
ST03D8A     2009/08/25  2009/08/25  08:59:53  TWUSR
ST04        2009/08/25  2009/08/25  08:59:52  TWUSR
ST05        2009/08/25  2009/08/25  08:59:53  TWUSR
ST06D9A3     2009/08/25  2009/08/25  08:59:50  TWUSR
ST06TI2     2009/08/25  2009/08/25  08:59:53  TWUSR
ST08D9A3     2009/08/25  2009/08/25  08:59:51  TWUSR
ST09D9A3     2009/08/25  2009/08/25  08:59:51  TWUSR
ST10D9A3     2009/08/25  2009/08/25  08:59:51  TWUSR
ST10TI2     2009/08/25  2009/08/25  08:59:53  TWUSR
ST11D9A3     2009/08/25  2009/08/25  08:59:51  TWUSR
ST11TI2     2009/08/25  2009/08/25  08:59:54  TWUSR
ST12D9A3     2009/08/25  2009/08/25  08:59:52  TWUSR
ST12TI2     2009/08/25  2009/08/25  08:59:54  TWUSR
ST13D9A3     2009/08/25  2009/08/25  08:59:52  TWUSR
ST13TI2     2009/08/25  2009/08/25  08:59:55  TWUSR
ST14D9A3     2009/08/25  2009/08/25  08:59:52  TWUSR
ST14TI2     2009/08/25  2009/08/25  08:59:55  TWUSR
ST22        2009/08/25  2009/08/25  08:59:55  TWUSR
```

Subsystem cloning job reference

This topic provides a list of the cloning job member names and the steps that they perform when executed.

Online cloning jobs

This table provides a list of the member names of the online cloning jobs that are generated and the commands that are run in the jobs.

| Member name | Command name | Details |
|--------------------|----------------------|---|
| ST01xxxx | DB2SETLOG SUSPEND | |
| ST02 | COPY | |
| ST03xxxx | DB2SETLOG RESUME | |
| ST04 | COPYCHECK | |
| ST05 | RENAME | |
| ST06xxxx | DB2UPDATE | For the primary member of a data sharing group and for all Db2 subsystems |
| ST07xxxx | DB2UPDATE | For the secondary target members of a data sharing group |
| ST08xxxx | DB2START | For the primary member of a data sharing group and for all Db2 subsystems |
| ST09xxxx | DB2START | For the secondary target members of a data sharing group |
| ST10xxxx | DB2FIX (Db2) | For the primary member of a data sharing group and for all Db2 subsystems |
| ST11xxxx | DB2STOP | For the primary member of a data sharing group and for all Db2 subsystems |
| ST12xxxx | DB2UPDATE DBD01ONLY | For the primary member of a data sharing group and for all Db2 subsystems |
| ST13xxxx | DB2START | For the primary member of a data sharing group and for all Db2 subsystems |
| ST14xxxx | DB2SQL | For the primary member of a data sharing group and for all Db2 subsystems |
| ST15xxxx | DB2FIX (application) | For the primary member of a data sharing group and for all Db2 subsystems |

Table 67. Online cloning job and member reference (continued)

| Member name | Command name | Details |
|-------------|-----------------------------|--|
| ST16xxxx | DB2SCHEMA-UPDATE (SIMULATE) | For the primary member of a data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST17xxxx | DB2SCHEMA-UPDATE | For the primary member of a data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST18xxxx | DB2STOP | For the primary member of a data sharing group and for all Db2 subsystems |
| ST19xxxx | DB2RBLDBSDS | For each surviving target Db2 member, only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST20xxxx | DB2LGRNXCLEAN | For the primary surviving target Db2 members, only if the value of Data sharing attributes of target is FEWER |
| ST21xxxx | DB2XCFCLEAN | For primary surviving target Db2 members, only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST22xxxx | DB2UTILXCLEAN | For the primary member of a data sharing group and for all Db2 subsystems |
| ST23xxxx | DB2START NORMAL | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST24xxxx | DB2STOP | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST25 | BCSCLEAN | |

Offline cloning jobs

This table provides a list of the member names of the offline cloning jobs that are generated and the commands that are run in the jobs.

| <i>Table 68. Offline cloning job and member reference</i> | | |
|---|---------------|--|
| Member name | Command name | Details |
| ST01xxxx | DB2STOP | |
| ST02 | COPY | |
| ST03xxxx | DB2START | |
| ST04 | COPYCHECK | |
| ST05 | RENAME | |
| ST06xxxx | DB2UPDATE | For the primary member of a data sharing group and for all Db2 subsystems |
| ST07xxxx | DB2UPDATE | For the secondary target members of a data sharing group |
| ST08xxxx | DB2RBLDBSDS | For each surviving target member and only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST09xxxx | DB2LGRNXCLEAN | For the primary surviving target member and only if the value of Data sharing attributes of target is FEWER |
| ST10xxxx | DB2START | For the primary member of a data sharing group, under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST11xxxx | DB2SQL | For the primary member of a data sharing group, under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |

Table 68. Offline cloning job and member reference (continued)

| Member name | Command name | Details |
|-------------|-----------------------------|--|
| ST12xxxx | DB2SCHEMA-UPDATE (SIMULATE) | For the primary member of a data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST13xxxx | DB2SCHEMA-UPDATE | For the primary member of a data sharing group and for all Db2 subsystems if the list of schema masks pairs is not empty |
| ST14xxxx | DB2STOP | For the primary member of a data sharing group, under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST15xxxx | DB2UTILXCLEAN | For the primary member of a data sharing group and for all Db2 subsystems |
| ST16xxxx | DB2START NORMAL | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST17xxxx | DB2STOP | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST25 | BCSCLEAN | |

Jobs for online cloning from a Db2 system level backup

This table provides a list of the member names of the jobs that are generated when online cloning from a system level backup, and the commands that are run in the jobs.

Table 69. Job and member reference for online cloning from a system level backup

| Member name | Command name | Details |
|-------------|---------------------|--|
| ST01 | DB2GETBACKINFO | |
| ST02 | BACKINFO-REFORMAT | |
| ST03 | COPY (PGM NONE) | |
| ST04 | COPY | |
| ST05 | COPYCHECK | |
| ST06 | CKZRNTGT | |
| ST07 | VOOPTIONS | |
| ST08 | RENAME | |
| ST09xxxx | DB2UPDATE | For the primary member of a data sharing group and for all Db2 subsystems |
| ST10xxxx | DB2UPDATE | For the secondary target members of a data sharing group |
| ST11xxxx | DB2ALTERBSDS | For all target members |
| ST12xxxx | DB2START | For the primary member of a data sharing group and for all Db2 subsystems |
| ST13xxxx | DB2START | For the secondary target members of a data sharing group, except for the last secondary member |
| ST14xxxx | DB2START | For last secondary member of a data sharing group |
| ST15xxxx | DB2STOP | For the secondary members of a data sharing group, except for the last secondary member |
| ST16xxxx | DB2FIX (Db2) | For the primary member of a data sharing group and for all Db2 subsystems |
| ST17xxxx | DB2STOP | For the primary member of a data sharing group and for all Db2 subsystems |
| ST18xxxx | DB2UPDATE DBD01ONLY | For the primary member of a data sharing group and for all Db2 subsystems |
| ST19xxxx | DB2START | For the primary member of a data sharing group and for all Db2 subsystems |
| ST20xxxx | DB2SQL | For the primary member of a data sharing group and for all Db2 subsystems |

Table 69. Job and member reference for online cloning from a system level backup (continued)

| Member name | Command name | Details |
|-------------|-----------------------------|--|
| ST21xxxx | DB2FIX (application) | For the primary member of a data sharing group and for all Db2 subsystems |
| ST22xxxx | DB2SCHEMA-UPDATE (SIMULATE) | For the primary member of a data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST23xxxx | DB2SCHEMA-UPDATE | For the primary member of a data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST24xxxx | DB2STOP | For the primary member of a data sharing group and for all Db2 subsystems |
| ST25xxxx | DB2RBLDBSDS | For each surviving target Db2 member, only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST26xxxx | DB2LGRNXCLEAN | For the primary surviving target Db2 members, only if the value of Data sharing attributes of target is FEWER |
| ST27xxxx | DB2XCFCLEAN | For primary surviving target Db2 members, only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST28xxxx | DB2UTILXCLEAN | For the primary member of a data sharing group and for all Db2 subsystems |
| ST29xxxx | DB2START NORMAL | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |

Table 69. Job and member reference for online cloning from a system level backup (continued)

| Member name | Command name | Details |
|-------------|--------------|--|
| ST30xxxx | DB2STOP | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST31 | BCSCLEAN | |

Jobs for online cloning from a Db2 system level backup using dump tapes as source

This table provides a list of the member names of the jobs that are generated when online cloning from a system level backup using dump tapes as a source, and the commands that are run in the jobs.

Table 70. Job and member reference for online cloning from a system level backup using dump tapes as source

| Member name | Command name | Details |
|-------------|-------------------------------|--|
| ST01 | DB2GETBACKINFO | |
| ST02 | RESTORE-FROM-DUMPTAPES | |
| ST02RR | RESTORE-FROM-DUMPTAPES(RERUN) | |
| ST03 | COPY (PGM NONE) | |
| ST08 | RENAME | |
| ST09xxxx | DB2UPDATE | For the primary member of a data sharing group and for all Db2 subsystems |
| ST10xxxx | DB2UPDATE | For the secondary target members of a data sharing group |
| ST11xxxx | DB2ALTERBSDS | For all target members |
| ST12xxxx | DB2START | For the primary member of a data sharing group and for all Db2 subsystems |
| ST13xxxx | DB2START | For the secondary target members of a data sharing group, except for the last secondary member |
| ST14xxxx | DB2START | For the last secondary target member of a data sharing group |
| ST15xxxx | DB2STOP | For secondary members of a data sharing group, except for the last secondary member |

Table 70. Job and member reference for online cloning from a system level backup using dump tapes as source (continued)

| Member name | Command name | Details |
|-------------|-----------------------------|---|
| ST16xxxx | DB2FIX (Db2) | For the primary member of a data sharing group and for all Db2 subsystems |
| ST17xxxx | DB2STOP | For the primary member of a data sharing group and for all Db2 subsystems |
| ST18xxxx | DB2UPDATE DBD01ONLY | For the primary member of a data sharing group and for all Db2 subsystems |
| ST19xxxx | DB2START | For the primary member of a data sharing group and for all Db2 subsystems |
| ST20xxxx | DB2SQL | For the primary member of a data sharing group and for all Db2 subsystems |
| ST21xxxx | DB2FIX (application) | For the primary member of a data sharing group and for all Db2 subsystems |
| ST22xxxx | DB2SCHEMA-UPDATE (SIMULATE) | For the primary member of a data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST23xxxx | DB2SCHEMA-UPDATE | For the primary member of data sharing group and for all Db2 subsystems, if the list of schema masks pairs is not empty |
| ST24xxxx | DB2STOP | For the primary member of a data sharing group and for all Db2 subsystems |
| ST25xxxx | DB2RBLDBSDS | For each surviving target Db2 member, only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST26xxxx | DB2LGRNXCLEAN | For the primary surviving target Db2 members, only if the value of Data sharing attributes of target is FEWER |
| ST27xxxx | DB2XCFCLEAN | For primary surviving target Db2 members, only if the value of Data sharing attributes of target is FEWER or NONDS |
| ST28xxxx | DB2UTILXCLEAN | For the primary member of a data sharing group and for all Db2 subsystems |

Table 70. Job and member reference for online cloning from a system level backup using dump tapes as source (continued)

| Member name | Command name | Details |
|-------------|-----------------|--|
| ST29xxxx | DB2START NORMAL | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST30xxxx | DB2STOP | Under one of the following conditions: <ul style="list-style-type: none"> • If the value of Data sharing attributes of target is SAME • If the value of Data sharing attributes of target is FEWER or NONDS and the members are surviving • For each target Db2 subsystem |
| ST29 | BCSCLEAN | |

Submit the jobs

After the subsystem cloning jobs have been generated, submit the jobs in order.

About this task

Jobs are built with members names of STxx, where xx is an indicator of the order in which the jobs should be submitted. Jobs with the same xx number can be submitted simultaneously. For instance, jobs ST06D9A3 and ST06D9B3 can be run at the same time.

Table space cloning

This section describes the basic steps for table space cloning using the ISPF interface.

Note: Be sure the DDs required for table space cloning have been created before attempting to build a table space cloning profile.

Start the ISPF interface and create a table space cloning profile. After the cloning profile has been created, the **Edit DB2 Tablespace Clone Profile** menu is displayed, as shown in the following figure:

```

                                Edit DB2 Tablespace Clone Profile
Option ==>

Creator . . . : TWUSR           Name . . . . : TEST
Share Option . : UPDATE        Description . .
Source SSID .  :               Target SSID .  :

1 Source and Target DB2 subsystems
2 Source job
3 Target job
4 Report job
5 TCPIP Server job
6 Source TCPIP Server job Required for cross LPAR log apply

```

Table space cloning steps summary

The general steps for using the ISPF interface to clone a Db2 table space or index space are described in this topic.

Procedure

1. Ensure the source and target subsystems have been added and properly configured under the Administrative Options option.
2. Create a table space cloning profile.
3. Select or verify the source and target subsystems.
4. Edit and verify the source job settings.
5. Edit and verify the target job settings.
6. Optionally, edit and verify the report job settings.
7. If your system configuration requires the TCP/IP server job and it is not currently active on the target subsystem, edit and verify the TCP/IP server job settings. Refer to [“TCP/IP server job overview \(optional\)”](#) on page 208 for detailed information about the TCP/IP server job.
8. If you are using LOG-APPLY or XML processing, and your source and target subsystems are on different LPARs, edit and verify the source TCP/IP server job settings. Refer to [“Using LOG-APPLY across multiple LPARs”](#) on page 294 for detailed information about the source TCP/IP server job.
9. Build the table space cloning profile to generate the jobs.
10. Submit the jobs in the correct order.

Select the source and target Db2 subsystems

The first step is to specify the source and target Db2 subsystems for the table space cloning profile.

About this task

On the **Edit DB2 Tablespace Clone Profile** menu, select option 1 (Source and Target Db2 subsystems). The **Source and Target DB2 subsystems** panel is displayed, as shown in the following figure:

```

Source and Target DB2 subsystems
Command ===>

Commands: S - Edit Source DB2 SSID  T - Edit Target DB2 SSID

Creator . . . . : TWUSR           Name . . . . . : TEST PROFILE
Share Option . . : UPDATE         Description . . :

Source SSID . . . . . (asterisk to select from list)
Use group attach name . . . . . NO (Yes/No)
IP-VERSION6 . . . . . NO (Yes/No)
SERVER-IP . . . . .
SERVER-PORT . . . . . 5099 (1-65535)
TCPIP-STC-NAME . . . . . TCPIP

Target SSID . . . . . (asterisk to select from list)
Use group attach name . . . . . NO (Yes/No)
IP-VERSION6 . . . . . NO (Yes/No)
SERVER-IP . . . . .
SERVER-PORT . . . . . 5099 (1-65535)
TCPIP-STC-NAME . . . . . TCPIP
LOCATION . . . . .
USERID . . . . .
PASSWORD . . . . .

```

Procedure

1. Enter a source SSID in the **Source SSID** field, or enter ***** to select a subsystem from the list of subsystems that have been configured using Administrator functions.
2. Enter a target SSID in the **Target SSID** field, or enter ***** to select a subsystem from the list of subsystems that have been configured using Administrator functions.
3. When finished, press PF3 (END) to save and exit.
The **Edit DB2 Tablespace Clone Profile** is displayed.

Edit the source job

If defaults were specified in the table space cloning default specifications panels under User Settings, a table space cloning profile can be created simply by verifying the command settings and specifying the objects for the LISTDEF command.

About this task

On the **Edit DB2 Tablespace Clone Profile** menu, select option 2 (Source job). The **Setup Source Job** menu is displayed, as shown in the following figure:

```

Setup Source Job
Option ===>

Creator . . . . : TWUSR           Name . . . . . : TEST PROFILE
Share Option . . : UPDATE         Description . . :
Source SSID . . : TGS2           Target SSID . . : TGT1

1 Job card and qualifiers
2 DD Specification
3 SET Command
4 COPY Command
5 HLQDDDF Command
6 XML Object Definition
7 LISTDEF Commands
8 Data Masking Commands

```

Procedure

1. On the **Setup Source Job** menu, enter 3 (SET Command).

The **DB2 tablespace clone SET Command** panel is displayed.

2. Verify or modify SET command settings as required.

3. Press PF3 (END) to exit.

4. On the **Setup Source Job** menu, enter 4 (COPY command).

The **DB2 tablespace clone COPY Command** panel is displayed.

5. Verify or modify COPY command settings as required.

6. Press PF3 (END) to exit.

7. On the **Setup Source Job** menu, enter 7 (LISTDEF Commands).

The **DB2 tablespace clone LISTDEF Commands** panel is displayed.

8. Specify the objects to be copied using this panel.

9. When finished, press PF3 (END) to exit, then press PF3 again.

The **Edit DB2 Tablespace Clone Profile** is displayed.

Verify the target job settings

The job cards and DDs for the target job may have already been specified in the user defaults; verify that the settings are correct.

About this task

On the **Edit DB2 Tablespace Clone Profile** menu, select option 3 (Target job). The **Setup Target Job** menu is displayed, as shown in the following figure:

```
Option  ===>                Setup Target Job
Creator   . . . : TWUSR        Name   . . . . : TEST PROFILE
Share Option . : UPDATE       Description . :
1 Job card
2 DD Specification
```

Select each menu option to verify settings.

Verify the report job settings

The report job is optional; it prints a report based on data in the target job runtime repository. The job cards and DDs for the report job may have already been specified in the user defaults; verify that the settings are correct.

About this task

On the **Edit DB2 Tablespace Clone Profile** menu, select option 4 (Report job). The **Setup Report Job** menu is displayed, as shown in the following figure:

```
Option  ===>                Setup Report Job
Creator   . . . : TWUSR        Name   . . . . : TEST PROFILE
Share Option . : UPDATE       Description . :
1 Job card
2 DD Specification
```

Select each menu option to verify settings.

Verify the TCP/IP server job settings

The TCPIP server job is optional; it facilitates communication between the source job and a target Db2 subsystem on a different z/OS system. The job cards and DDs for the TCPIP job may have already been specified in the user defaults; verify that the settings are correct.

About this task

On the **Edit DB2 Tablespace Clone Profile** menu, select option 5 (TCPIP job). The **Setup TCPIP Server Job** menu is displayed, as shown in the following figure:

```
Option  ===>                Setup TCPIP Server Job
Creator  . . . : TWUSR          Name . . . . : TEST PROFILE
Share Option . : UPDATE        Description . :
1 Job card
2 DD Specification
3 SET Command
```

Select each menu option to verify settings.

Verify the source TCP/IP server job settings

The source TCP/IP server job is required only when you are using LOG-APPLY or XML processing and you are cloning across multiple LPARs. It facilitates communication between the target job and source Db2 subsystem.

About this task

On the **Edit DB2 Tablespace Clone Profile** menu, select option 6 (Source TCPIP job). The **Setup Source TCPIP Server Job** menu is displayed, as shown in the following figure:

```
Option  ===>                Setup Source TCPIP Server Job
Creator  . . . : TWUSR          Name . . . . : TEST PROFILE
Share Option . : UPDATE        Description . :
1 Job card
2 DD Specification
3 SET Command
```

Select each menu option to verify settings.

Build the table space cloning jobs from a profile

Once the profile has been created, build the profile to produce the table space cloning jobs.

Before you begin

Be sure the DDs required for table space cloning have been created before attempting to build a table space cloning profile.

About this task

Begin building the profile on the **DB2 Tablespace Clone Profile Display**, as shown in the following figure:

```

DB2 Tablespace Clone Profile Display
Command ==>                               Scroll ==> CSR
Commands:      C - Create
Line Commands: B - Build D - Delete E - Edit R - Rename V - View C - Copy

Profile Like . . . *
Creator Like . . . TWUSR*

Row 1 of 1
Cmd Name      Creator  Share Option Description
TEST PROFILE  TWUSR   UPDATE

```

Procedure

1. On the **DB2 Tablespace Clone Profile Display**, enter B next to the profile that you want to build.
2. On the **Build DB2 tablespace clone jobs** menu, enter 1 to generate source and target jobs and press Enter.
The **Generate Source and Target Jobs** panel is displayed.
3. Specify the data set and member names for the source and target jobs.
4. If desired, select one or more processing options.
5. Press PF3 (END) to continue.
If you specified to review the jobs, they are displayed in an ISPF edit session.
6. Press PF3 (END) until you return to the **Build DB2 tablespace clone jobs** menu.
7. If you want to generate the report job:
 - a) Enter 2 and press Enter.
The **Generate Report Job** panel is displayed.
 - b) Specify the data set and member name for the report job.
 - c) If desired, select one or more processing options.
 - d) Press PF3 (END) to continue.
If you specified to review the job, it is displayed in an ISPF edit session.
 - e) Press PF3 (END) until you return to the **Build DB2 tablespace clone jobs** menu.
8. If you want to generate the TCP/IP server job:
 - a) Enter 3 and press Enter.
The **Generate TCPIP Server Job** panel is displayed.
 - b) Specify the data set and member name for the TCP/IP server job.
 - c) If desired, select one or more processing options.
 - d) Press PF3 (END) to continue.
If you specified to review the job, it is displayed in an ISPF edit session.
 - e) Press PF3 (END) until you return to the **Build DB2 tablespace clone jobs** menu.
9. If you want to generate the source TCP/IP server job (required only when you are using LOG-APPLY or XML processing and cloning across multiple LPARS):
 - a) Enter 4 and press Enter.
The **Generate Source TCPIP Server Job** panel is displayed.
 - b) Specify the data set and member name for the source TCP/IP server job.
 - c) If desired, select one or more processing options.
 - d) Press PF3 (END) to continue.
If you specified to review the job, it is displayed in an ISPF edit session.
 - e) Press PF3 (END) until you return to the **Build DB2 tablespace clone jobs** menu.

Results

The jobs have been generated and are in the data sets specified. You can now submit the table space cloning jobs to perform the cloning.

Submit the jobs

After the jobs have been generated, submit the jobs in the correct order.

Procedure

1. Run the optional TCP/IP server job if required.
2. Run the optional source TCP/IP server job, if required.
3. Run the source job.
4. Run the target job.
5. Run the optional report job.

Results

The table space cloning is now complete.

Table space cloning across multiple profiles

You might need to spread objects to be cloned across several profiles, usually because there are many objects to be cloned. You can choose to apply LOG-APPLY settings to multiple profiles on the **DB2 Tablespace Clone Profile Display**. From the same panel, you can create synchronization jobs to clone the objects spread across multiple profiles to the same TO_LOGPOINT.

Making LOG-APPLY setting changes across multiple profiles

Use one of the following methods to select multiple profiles on the **DB2 Tablespace Clone Profile Display** and make changes to LOG-APPLY settings that apply to all of the selected profiles.

Method 1:

1. On the **DB2 Tablespace Clone Profile Display**, select two or more profiles with the L line command.
2. Press Enter.
3. Enter desired LOG-APPLY settings on the **DB2 tablespace LOG-APPLY Command** panel. These settings apply to all the profiles that were selected on the **DB2 Tablespace Clone Profile Display**.

Method 2:

1. On the **DB2 Tablespace Clone Profile Display**, use the **Profile Like** and **Creator Like** fields to filter the list of profiles to display only the profiles to which you want to apply LOG-APPLY settings.
2. Enter the L command in the **Option** line.
3. Enter desired LOG-APPLY settings on the **DB2 tablespace LOG-APPLY Command** panel. These settings apply to all the profiles that were listed on the **DB2 Tablespace Clone Profile Display**.

Generating jobs to synchronize the log point across multiple profiles

Use one of the following methods to generate TO-LOGPOINT synchronization batch JCL jobs for all of the displayed or selected profiles. The synchronization jobs allow objects that are spread over multiple profiles to be cloned to the same log point.

If the table space cloning profiles contain different source SSIDs, Db2 Cloning Tool generates different TO-LOGPOINT synchronization jobs, one for each SSID. If the profiles contain the same source SSID, one job is generated.

Method 1:

1. On the **DB2 Tablespace Clone Profile Display**, select two or more profiles with the S line command.

2. Press Enter.
3. On the **Build TO-LOGPOINT Synchronization Jobs** panel, specify the data set and member name prefix for the synchronization JCL jobs.
4. Press PF3.
5. If any profiles cannot be processed, they are listed on the **DB2 Tablespace Clone Profile Display**. Press PF3 to continue.
6. A message is displayed stating that the JCL was successfully generated.

Method 2:

1. On the **DB2 Tablespace Clone Profile Display**, use the **Profile Like** and **Creator Like** fields to filter the profile list to display only the profiles that contain objects for which you want a synchronized log point.
2. Enter the S command in the **Option** line.
3. On the **Build TO-LOGPOINT Synchronization Jobs** panel, specify the data set and member name prefix for the synchronization JCL jobs.
4. Press PF3.
5. If any profiles cannot be processed, they are listed on the **DB2 Tablespace Clone Profile Display**. Press PF3 to continue.
6. A message is displayed stating that the JCL was successfully generated.

For both methods, generate your source and target jobs first, then generate the synchronization jobs. Run the jobs as follows, assuming there are n source and target jobs:

1. Run the first to the n source jobs sequentially.
2. Run the TO-LOGPOINT synchronization job or jobs.
3. Run the first to the n target jobs sequentially.

Chapter 24. Table space cloning using z/OSMF REST services

Table space cloning can be accomplished using a z/OSMF Representational State Transfer (REST) API. A sample workflow for table space cloning is provided in the SCKZJCL library. This solution provides you with a method to efficiently clone objects using DevOps provisioning via REST calls.

Restriction: To use the z/OSMF workflow, table space cloning must be accomplished using CAF for the connection to the target Db2 subsystem.

The samples that are provided include templates, configuration, and input files, and instructions on how to modify the files and use these files in a workflow. The samples that are provided are intended to interface with the z/OSMF web interface and with z/OSMF REST APIs. More information about z/OSMF REST APIs and the web browser-based interface can be found in the *IBM z/OSMF Configuration Guide* and the *IBM z/OSMF Programming Guide* in IBM Knowledge Center.

Sample members and descriptions

The SCKZJCL library provides sample members to reproduce a table space cloning scenario using the z/OSMF REST API.

A brief description of the provided members follows.

CKZWFXML

This member contains the workflow definition file and is written in XML. The sample workflow definition contains steps to run a simple table space cloning scenario when there is a CAF connection to the target Db2 subsystem and shared DASD between the source and target systems. The workflow contains five steps:

1. Extraction of multi-line variables (for LISTDEF, OBJECT-TRANSLATE, and DDL-ATTRIBUTE-CHANGE values) that may require additional checks, depending on other variables' values. The extracted values are written to an additional file that is created in a user-specified directory, with the name that is specified in the **propertyFile** variable in the variable input file. This property file is later used to substitute variables into the job templates.
2. Parameters setup. All specified variables are checked for conflicts and to ensure that all required values are specified. Warning and/or error messages may be issued in the step output. Errors will stop the workflow from proceeding.
3. Customize variables. This step constructs the variables that are only present when they are specified or when they are based on other parameter values.
4. Run the table space cloning source job.
5. Run the table space cloning target job.

CKZWFVAR

This member is the variable input file and contains a list of variables and their values in the format 'variable = value'. Variables are specified instead of values in the delivered variables file sample (in format &&var). These variables should be changed to the desired values before workflow creation. Variables that can be specified for the workflow sample are listed in [“Variable input file parameters” on page 358](#).

CKZWFSRJ

This member contains the Db2 Cloning Tool Table Space Cloning source job JCL template.

CKZWFTRJ

This member contains the Db2 Cloning Tool Table Space Cloning target job JCL template.

CKZWFPCPY

This member contains JCL to prepare a USS environment for workflow creation. Run this job first to configure the samples on USS. The job contains the following steps:

1. Creation of a USS directory to contain the workflow files.
2. Copying of files from the Db2 Cloning Tool sample library to the created USS directory.
3. Trim trailing spaces from files on USS after copying from the Db2 Cloning Tool sample library and set required authorities for workflow files. The name of the sample library and the USS directory are specified by the variables '&&CKZ.SAMPLE.LIBRARY' and '&&your/directory/path', and should be changed to the appropriate values for your site.

Note: Ensure that all of the directories in the specified path allow read access for all users of this workflow. If read access must be granted, after following the steps above, run the **chmod 644** command on the appropriate directories.

Variable input file parameters

The variables in the CKZWFVAR file should be modified before workflow creation.

The following table describes the variables and their possible values, and provides additional information that may be needed when modifying the variable input file.

| Variable | Variable value | Acceptable values | Number of lines allowed |
|--------------|------------------|--|-------------------------|
| srcSCKZLOAD | &&SRC_CKZLOAD | Db2 Cloning Tool LOAD library name, up to 44 characters. | 1 |
| tgtSCKZLOAD | &&TRG_CKZLOAD | Db2 Cloning Tool LOAD library name, up to 44 characters. | 1 |
| srcCKZINI | &&SRCINI | Db2 Cloning Tool PARM library and member name, up to 54 characters. | 1 |
| tgtCKZINI | &&TRGINI | Db2 Cloning Tool PARM library and member name, up to 54 characters. | 1 |
| CKZPlan | &&CKZPLAN | Db2 Cloning Tool plan name, up to 8 characters. | 1 |
| syncdb2dsn | &&SYNCDB2_DDN | Db2 Cloning Tool SYNCDB2 work data set name without member name, up to 44 characters. | 1 |
| ddloutdsn | &&DDL_DDN | Db2 Cloning Tool DDL work data set name without member name, up to 44 characters. | 1 |
| logapplydsn | &&LOGAP_DDN | Db2 Cloning Tool target LOG-APPLY work data set name without member name, up to 44 characters. | 1 |
| srcTCPIPport | &&SRC_TCPIP_PORT | Db2 Cloning Tool Source TCP/IP server job port, is not used, can be changed to default value 5099, up to 5 characters. | 1 |
| tgtTCPIPport | &&TRG_TCPIP_PORT | Db2 Cloning Tool Target TCP/IP server job port, is not used, can be changed to default value 5099, up to 5 characters. | 1 |
| srcLPAR | &&SRCLPAR | Name of the LPAR where the source Db2 subsystem runs, up to 4 characters. | 1 |

| <i>Table 71. Workflow sample variables (continued)</i> | | | |
|--|-----------------------|---|--------------------------------|
| Variable | Variable value | Acceptable values | Number of lines allowed |
| tgtLPAR | &&TRGLPAR | Name of the LPAR where the target Db2 subsystem runs, up to 4 characters. | 1 |
| srcdb2 | &&SRCDDB2 | Source Db2 subsystem name, up to 4 characters. | 1 |
| tgtdb2 | &&TRGDB2 | Target Db2 subsystem name, up to 4 characters. | 1 |
| srcSDSNLOAD | &&SRC_SDSNLOAD | Db2 LOAD library for the source Db2 subsystem, up to 44 characters. | 1 |
| tgtSDSNLOAD | &&TRG_SDSNLOAD | Db2 LOAD library for the target Db2 subsystem, up to 44 characters. | 1 |
| srcSDSNEXIT | &&SDSNEXIT | Db2 EXIT library for the source Db2 subsystem, up to 44 characters. | 1 |
| srcZPARAM | &&SRCZPARAM | Name of the source Db2 subsystem ZPARAM member, up to 8 characters. | 1 |
| srcBSDS1 | &&BSDS01 | First source Db2 subsystem BSDS library, up to 44 characters. | 1 |
| srcBSDS2 | &&BSDS02 | Second source Db2 subsystem BSDS library, up to 44 characters. | 1 |
| tgtVCAT | &&TRGVCAT | Target data sets VCAT, up to 8 characters. | 1 |
| dsn_Member | &&MMBR_NAME | Member name for all work Db2 Cloning Tool data sets that are specified earlier, up to 8 characters. | 1 |
| simulate | &&SIM | Y N, value of the SIMULATE table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| dataMoverPgm | &&DATA_MOVER | ADRDUSSU NONE are currently the only supported values for DATA-MOVER PGM table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| fastrep | &&FASTREP | PREF REQ NONE for FASTREP table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |

Table 71. Workflow sample variables (continued)

| Variable | Variable value | Acceptable values | Number of lines allowed |
|------------------|-----------------|---|-------------------------|
| fctopprcprimary | &&PRESMIR | PRESMIRREQ PRESMIRPREF PRESMIRNONE for FCTOPPRCPRI­MARY table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| remoteConnType | &&RCTYP | Currently, only C is supported for CAF connection, for REMOTE-CONNECT-TYPE table space cloning parameter. | 1 |
| fuzzycopy | &&FUZZY | Y N for FUZZY-COPY table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| maxSubtasks | &&SUBTSK | From 1 to 99 for MAX-SUBTASKS table space cloning parameter (see “SET command and keyword definitions” on page 568 for parameter values explanation). | 1 |
| LAenable | &&LOGAP_ENABLED | Y N for LA-ENABLED table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| minilogHLQ | &&MINILOG-HLQ | HLQ for LOG APPLY MINILOG work data sets, up to 35 characters (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| spacesPerML | &&MINILOG_SPC | From 1 to 99 for SPACES-PER-MINILOG table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| LALogpoint | &&LOGPOINT | TO_CURRENT (default) TO_LOGPOINT X'byte_string' TO_QUIESCE TO_TIMESTAMP <i>timestamp</i> for END-POINT table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| intLAdatasharing | - | For internal use only. Do not change. | |

| <i>Table 71. Workflow sample variables (continued)</i> | | | |
|--|---|--|--------------------------------|
| Variable | Variable value | Acceptable values | Number of lines allowed |
| LAdatasharing | *** | DATA-SHARING-MEMBERS table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). Use ‘***’ if the source subsystem is not in a data sharing group, otherwise, see the topic “Considerations for multi-line variable values” on page 363 for information on how to specify a value for this parameter. | Multiple lines are allowed |
| rtsEnable | Y | Y N for RTS-COPY-ENABLE table space cloning parameter, Y is recommended (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| rtsHLQ | &&RTSHLQ | HLQ for RTS work data sets, up to 35 characters (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| propertyFile | &&USS_PATH_TO_PROPERTY_FILE | USS path (including file name) to workflow work file, is recommended to be the same path as the workflow definition file with extension .txt. The file can be any name that you choose. | 1 |
| intListdef | - | For internal use. Do not change. | |
| listdef | \ &&LDF_HEADER \\n\\ \ &&LDF_RULE01 \\n\\ \ &&LDF_RULE02 \\n\\ \ &&LDF_RULE03 \\n\\ \ \\ | LISTDEF statements. See the topic “Considerations for multi-line variable values” on page 363 for information on how to specify a value for this parameter. | Multiple lines are allowed. |
| inclAllRI | &&LDF_RI | Y N for INCLUDE-ALL-RI table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| explodeObj | &&EXPLODE | Y N for PROCESS-DDL EXPLODE-OBJECTS table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |

Table 71. Workflow sample variables (continued)

| Variable | Variable value | Acceptable values | Number of lines allowed |
|----------------|---------------------------------|--|-----------------------------|
| supprRIconst | &&SUPPRESS_RI | Y N for PROCESS-DDL SUPPRESS-RI-CONSTRAINTS table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| ddlproc | &&DDL_PROC_TYPE | Y G X A N for PROCESS-DDL PROCESS-TYPE table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| useDDLsqlid | *** | PROCESS-DDL PROCESS-TYPE table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). Use ‘***’ to use catalog SQLID, otherwise specify the desired SQLID to make the DDL generator generate ‘SET CURRENT SQLID’ statements. | 1 |
| intUseDDLsqlid | - | For internal use. Do not change. | |
| commitFreq | &&COMMIT_FREQ | From 1 to 1000 for PROCESS-DDL COMMIT-FREQUENCY table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |
| intDdlattr | - | For internal use. Do not change. | |
| ddlattr | \ &&DDLATTR_RULE01 - \\n\\ \ | PROCESS-DDL DDL-ATTRIBUTE-CHANGE table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). See the topic “Considerations for multi-line variable values” on page 363 for information on how to specify a value for this parameter. Note: at least one value must be specified. | Multiple lines are allowed. |
| intObjxlat | - | For internal use. Do not change. | |

Table 71. Workflow sample variables (continued)

| Variable | Variable value | Acceptable values | Number of lines allowed |
|--------------|---|---|-----------------------------|
| objxlat | \ &&OBX_RULE01 - \\n\\ \ &&OBX_RULE02 - \\n\\ \ &&OBX_RULE03 - \\n\\ \ | OBJECT-TRANSLATE table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). See the topic “Considerations for multi-line variable values” on page 363 for information on how to specify a value for this parameter. Note: At least one rule is required. If source SSID = target SSID, rules for DATABASE and CREATOR translation should be specified. If source SSID is not equal to target SSID, VCAT translation should be specified. | Multiple lines are allowed. |
| xmlDOCIDfile | *** | UPDATE-DOCID-JCL-DSN table space cloning parameter (see “SET command and keyword definitions” on page 568 for parameter values explanation). If XML spaces will be cloned, the DSN name should be specified here (up to 44 characters), otherwise use ‘***’. | 1 |
| tgtStart | &&START_TRG_SPACE | Y N for AUTO-START-TARGET-SPACE table space cloning parameter (see “COPY command and keyword definitions” on page 516 for parameter values explanation). | 1 |

Considerations for multi-line variable values

Variables that can span multiple lines in the variable input file follow these continuation rules.

LISTDEF value

Set the line break with a backslash (\) on the first line. Each successive line for LISTDEF should terminate with the following string:

```
\\n\\ \  

```

Example 1:

```
listdef = \  
INCLUDE TABLESPACES DATABASE DB01 \\n\\ \  

```

Example 2:

```
listdef = \  
INCLUDE TABLESPACES \\n\\ \  
DATABASE \\n\\ \  
DB01 \\n\\ \  

```

```
INCLUDE INDEXSPACES \\n\\ \  
DATABASE DB01 \\n\\ \  

```

DATA-SHARING-MEMBERS value

Set the line break with a backslash (\) on the first line. Each successive line for DATA-SHARING-MEMBERS (except for the last one) should terminate with the following string:

```
- \\n\\ \  

```

For the last line, the comma (,) may be omitted.

Example:

```
LAdatasharing = \  
1,ssid1,zparm1, - \\n\\ \  
2,ssid2,zparm2, - \\n\\ \  
3,ssid3,zparm3, - \\n\\ \  
4,ssid4,zparm4 - \\n\\ \  

```

DDL-ATTRIBUTE-CHANGE value

Set the line break with a backslash (\) on the first line. Each successive line for a DDL-ATTRIBUTE-CHANGE rule should terminate with the following string:

```
- \\n\\ \  

```

Syntax for DDL-ATTRIBUTE-CHANGE rules is described in the topic [“COPY command and keyword definitions”](#) on page 516.

Example:

```
ddlattr = \  
STOGROUP,,SYSDEFLT,,, - \\n\\ \  

```

OBJECT-TRANSLATE value

Set the line break with a backslash (\) on the first line. Each successive line for an OBJECT-TRANSLATE rule should terminate with the following string:

```
- \\n\\ \  

```

Syntax for OBJECT-TRANSLATE rules is described in the topic [“COPY command and keyword definitions”](#) on page 516.

Example:

```
objxlat = \  
DATABASE,DBSRC%,DBTRG% - \\n\\ \  
CREATOR,CRSRC%,CRTRG% - \\n\\ \  
VCAT,VCATSRC,VCATTRG - \\n\\ \  

```

Considerations for multi-line variable values in a JSON request body

Variables that can span multiple lines in the JCL and that are passed through the JSON request body follow these continuation rules.

LISTDEF value

Variable values are set in one long string. Set the line breaks with a backslash and "n" (\n).

Example:

```
{"name": "listdef", "value": "LISTDEF CKZSAMP \nINCLUDE TABLESPACES \nDATABASE SRCDB01 \nRI ALL"}
```

DATA-SHARING-MEMBERS value

Variable values are set in one long string. Each set of values for DATA-SHARING-MEMBERS should terminate with the following string:

```
- \\n\\
```

Include a space before the hyphen.

Example:

```
{"name": "LAdatasharing", "value": "1,MBR1,MBR1PARM, - \\n\\ 2,MBR2,MBR2PARM - \\n\\ "}
```

DDL-ATTRIBUTE-CHANGE value

Variable values are set in one long string. Each set of values for a DDL-ATTRIBUTE-CHANGE rule should terminate with the following string:

```
-\\n
```

Include a space before the hyphen. Syntax for DDL-ATTRIBUTE-CHANGE rules is described in the topic [“COPY command and keyword definitions”](#) on page 516.

Example:

```
{"name": "ddlattr", "value": "STOGROUP,,SYSDEFLT,, -\\nVCAT,,TRGVCAT,,,"}
```

OBJECT-TRANSLATE value

Variable values are set in one long string. Set the line breaks with a backslash and "n" (\n). Include a space before the hyphen.

Syntax for OBJECT-TRANSLATE rules is described in the topic [“COPY command and keyword definitions”](#) on page 516.

Example:

```
{"name": "objxlat", "value": "DATABASE,SRCDB*,TRGDB* - \\nCREATOR,CRS*,CRT* - \\nVCAT,SRCVCAT,TRGVCAT -"}
```

Configuring and running the workflow

Follow these instructions to copy the provided samples from the SCKZJCL library to USS, and customize and run the workflow.

Procedure

1. Run the CKZWFPCPY job to copy the workflow files to USS.
The name of sample library and USS directory are specified by the variables ‘&&CKZ.SAMPLE.LIBRARY’ and ‘&&your/directory/path’, and should be changed to appropriate values.
2. Edit the variable input file CKZWFVAR to set values for cloning.
Refer to [“Variable input file parameters”](#) on page 358 for information about the variables.
3. Using either the z/OSMF web-based interface or REST API, run the workflow. Refer to [“Running the workflow via REST services”](#) on page 366 or [“Running the workflow via the z/OSMF web browser-based interface”](#) on page 368 for more information.

4. Review the job output to ensure that the cloning was successful.

Job output is available in z/OSMF under the **Status** tab of the workflow step, and in your job output system (SDSF). Output in either of these locations can be used to determine what happened during workflow execution and for initial debugging.

What to do next

Once an instance of the workflow has been run, it is marked complete, and should not be run again.

Running the workflow via REST services

After modifying the variable input file and workflow definition file, you can run the workflow via the z/OSMF workflow services, which are implemented via REST services APIs.

About this task

The use of the z/OSMF workflow via REST is a two-step process: creating a workflow and starting a workflow.

Procedure

1. To create a workflow, use a POST request.

The following request contains an example of overriding variables in the POST request body.

```
POST /zosmf/workflow/rest/1.0/workflows HTTP/1.1
Host: lpar:11443
Content-Type: application/json
Authorization: Basic <Generated value>
{
  "workflowName": "Workflow name",
  "workflowDefinitionFile": "path_to_workflow_definition_file",
  "variableInputFile": "path_to_variable_input_file",
  "system": "lpar",
  "owner": "owner",
  "jobStatement": " //jobname JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),REGION=0M, \n//
USER=userid,NOTIFY=userid \n/*",
  "variables":
  [
    { "name": "fuzzycopy", "value": "Y" },
    { "name": "LAenable", "value": "Y" },
    { "name": "LAdatasharing", "value": "1,MBR1,MBR1PARM, - \\\n\\ 2,MBR2,MBR2PARM - \\\n\\ " },
    { "name": "minilogHLQ", "value": "CKZDEMO.MINILOG" },
    { "name": "dsn_Member", "value": "CKZSAMP" },
    { "name": "ddlproc", "value": "Y" },
    { "name": "listdef", "value": "LISTDEF CKZSAMP \nINCLUDE TABLESPACES \nDATABASE SRCDB01 \nRI ALL" },
    { "name": "objxlat", "value": "DATABASE, SRCDB*, TRGDB* - \nCREATOR, CRS*, CRT* - \nVCAT, SRCVCAT, TRGVCAT" },
    - " } ,
    { "name": "ddlattr", "value": "STOGROUP, ,SYSDEFLT, , , -" },
    { "name": "useDDLsqlid", "value": "***" },
    { "name": "propertyFile", "value": "path_to_property_file_on_uss" },
    { "name": "xmlDOCIDfile", "value": "UID.CKZ.DOCID" },
    { "name": "src1par", "value": "SLPR" },
    { "name": "srcdb2", "value": "SDB2" },
    { "name": "srcSDSNLOAD", "value": "DB2VERS.SDSNLOAD" },
    { "name": "srcSDSNEXIT", "value": "SDB2.SDSNEXIT" },
    { "name": "srcBSDS1", "value": "SDB2.BSDS01" },
    { "name": "srcBSDS2", "value": "SDB2.BSDS02" },
    { "name": "srcZPARM", "value": "MBR1PARM" },
    { "name": "tgtlpar", "value": "TLPR" },
    { "name": "tgtdb2", "value": "TDB2" },
    { "name": "tgtVCAT", "value": "TRGVCAT" },
    { "name": "tgtStart", "value": "Y" },
    { "name": "tgtSDSNLOAD", "value": "DB2VERS.SDSNLOAD" }
  ]
}
```

This example shows most of the variables defined in the request text. However, it is not necessary to define these variables via the request text if these variables are already defined in the variable input file. All of the variables' values are taken from the variable input file, except for variables that are overridden in the "variables" group of the POST request.

Instead of deleting a variable that you will not use for this workflow, you can rename the variable with a name that does not exist in the workflow definition XML file. The modified variable is not used for this workflow, but remains in the JSON payload as a reminder to re-assess the variable when the payload is copied for reuse. For example, for a request where the source is not a data sharing group, change "LAdatasharing" to "X-LAdatasharing".

All REST calls require authorization with the basic authentication method. <Generated value> is code that is generated by the user name and the password. The workflow accessType default value is Public. All workflow information is available to all users.

Note: It is not recommended to change the workflow assignToOwner value to false; REST services do not allow for changing an assignee for workflow steps.

After the POST request is submitted, one of the following is returned:

- If the request was successful, status = 201 (Created) is returned and the JSON response structure is:

```
{
  "vendor": "Vendor",
  "workflowKey": "key-value",
  "workflowVersion": "1.0",
  "workflowDescription": "Execute Db2 Cloning Tool Application Cloning \n",
  "workflowID": "CTappCloning"
}
```

All of the operations that follow for the created workflow are enabled via the returned z/OSMF **workflowKey key-value**, which is a unique key generated by z/OSMF.

- If a workflow with the specified workflowName already exists, status = 400[®] is returned and the JSON response structure is:

```
{
  "messageText": "IZUWF0006E: A workflow with the name \" Workflow name\" already exists.
                Enter a unique workflow name.",
  "messageID": "IZUWF0006E"
}
```

- If there is a problem with access to files that are specified in the workflow definition file, status = 400 is returned and the JSON response structure is:

```
{
  "messageText": "IZUWF0101E: Workflow definition file \"
path_to_workflow_definition_file \"
                was either not found or cannot be accessed.",
  "messageID": "IZUWF0101E"
}
```

2. To start a workflow, use a PUT request.

The following request contains an example of starting a workflow. *key-value* in the request should be the workflowKey value that was returned in response for POST request used for workflow creation.

```
PUT /zosmf/workflow/rest/1.0/workflows/key-value/operations/start HTTP/1.1
Host: lpar:11443
Authorization: Basic <Generated value>
Cache-Control: no-cache
```

This request is used to start an automated workflow; that is, a workflow with at least one step that can be performed automatically. If a workflow with the specified key exists, the workflow starts with first step that was not previously run (that is, in "Ready" status). Status = 202 is returned. No information about the result of the execution is provided. To monitor workflow status, GET requests can be used. For example:

```
GET /zosmf/workflow/rest/1.0/workflows/key-value HTTP/1.1
Host: lpar:11443
Authorization: Basic <Generated value>
```

For more information, see [z/OS jobs REST interface](#) in the IBM z/OS Management Facility online documentation.

What to do next

You can cancel or delete a workflow using the following requests:

- To cancel a workflow, use a PUT request. *key-value* in the request is the workflowKey value that was returned in the response for the POST request that was used for workflow creation. The following request shows an example of canceling a workflow.

```
PUT /zosmf/workflow/rest/1.0/workflows/key-value/operations/cancel HTTP/1.1
Host: lpar:11443
Authorization: Basic <Generated value>
Cache-Control: no-cache
```

A workflow cannot be canceled if its statusName = "automation-in-progress" or "cancelled". In that case, status = 409 (Conflict) is returned and the JSON response structure is:

```
{
  "messageText": "IZUWF0147E: Automation processing cannot proceed for workflow
  \"workflow_key\".
  To be eligible for automation, a workflow must be either In Progress or Complete.",
  "messageID": "IZUWF0147E"
}
```

If statusName = "in-progress" or statusName = "complete", status = 200 (OK) is returned.

If a cancel request was successful and statusName = "canceled", the workflow cannot be started again.

- To delete a workflow, use a DELETE request. *key-value* in the request is the workflowKey value that was returned in the response for the POST request that was used for workflow creation. The following request shows an example of deleting a workflow.

```
DELETE /zosmf/workflow/rest/1.0/workflows/key-value HTTP/1.1
Host: lpar:11443
Authorization: Basic <Generated value>
Connection: close
```

A workflow cannot be deleted if statusName="automation-in-progress". Status = 409 (Conflict) is returned. If a delete request is successful, status = 204 (No content) is returned.

Running the workflow via the z/OSMF web browser-based interface

After modifying the variable input file and workflow definition file, you can run the workflow via the z/OSMF web browser-based user interface.

About this task

The interface is not shown in this topic. For more information about working with the workflow through the z/OSMF web browser-based interface, see *IBM z/OSMF Configuration Guide* and the *IBM z/OSMF Programming Guide* on IBM Knowledge Center.

Procedure

1. Create a workflow by specifying the USS paths for the workflow definition file and the variable input file.
2. Run the workflow.
3. If any of the first three steps finishes with an error, review the step output. The variable input file might require changes. If a different problem occurs, contact IBM Software Support.

Chapter 25. Cloning table spaces from Db2 Administration Tool

Db2 Cloning Tool integrates with IBM Db2 Administration Tool for z/OS (also referred to as Db2 Admin). Use Db2 Admin to select the table spaces to clone. Then to clone the spaces, invoke Db2 Cloning Tool directly from the Db2 Admin interface.

Before you begin

Ensure that the Db2 Cloning Tool customization tasks that are described in [Chapter 4, “Customizing Db2 Cloning Tool,”](#) on page 77 and [Chapter 5, “After customizing Db2 Cloning Tool using Tools Customizer,”](#) on page 97 have been completed.

Procedure

1. Run the Db2 Admin CLIST.
2. Select table spaces to clone using the **Databases** panel, the **Table Spaces** panel, or the **Tables, Views, and Aliases** panel.
3. Type the **CT** command on the command line and press Enter.
All objects listed on the displayed panel are selected for cloning.
4. Optional: On the **Clone Tables** panel, refine the list of spaces to be cloned by using the available Db2 Admin commands.
5. On the **Clone Tables** panel, type **CONTINUE** on the command line and press Enter.
6. Create a new cloning profile or select an existing Db2 Cloning Tool cloning profile.
 - To create a new profile, complete the following steps:
 - a. On the **Specify clone profile** window, type **1** in the **Select an action** field, and press Enter.
 - b. On the **Enter New DB2 Tablespace Clone Profile Options** window, enter a new profile name and share option. Press Enter.
 - To use an existing profile, complete the following steps:
 - a. On the **Specify clone profile** window, type **2** in the **Select an action** field, and press Enter.
 - b. On the **DB2 Tablespace Clone Profile Selection** panel, select a profile, and press Enter.
 - c. You are prompted to add the table spaces that you selected using Db2 Admin to the existing profile, or replace the existing table spaces in the profile with those you selected using Db2 Admin. Press Enter.
7. On the **Edit DB2 Tablespace Clone Profile** panel, type **1** to edit the source job, and press Enter.
8. On the **Setup Source Job** panel, type **4** to edit the COPY command information, and press Enter.
9. On the **DB2 tablespace clone COPY Command** panel, enter the target subsystem in the **TARGET-DB2 SSID** field, or if using an existing profile, verify that the correct target subsystem has been specified. If the target has not been specified, type the target subsystem name in the field. To select the subsystem from a list, type ***** in the field, and press Enter. If the subsystem does not exist:
 - a) Type the target subsystem name in the **TARGET-DB2 SSID** field.
 - b) Type **I** in the command line and press Enter.
 - c) Specify the required information for the target subsystem that is described in the topic [“Configuring Db2 subsystems”](#) on page 324.
10. On the **Setup Source Job** panel, type **7** to review LISTDEF command information, and press Enter.
11. On the **DB2 Tablespace Clone LISTDEF Commands** panel, review the list of table spaces that were selected using Db2 Admin and modify it if required.

12. Press PF3 until the **Edit DB2 Tablespace Clone Profile** panel is displayed.
13. Continue with table space cloning process, following the steps starting with the topic [“Edit the source job”](#) on page 351.

Chapter 26. Db2 Cloning Tool Subsystem Cloning commands

All Db2 Cloning Tool Subsystem Cloning commands are invoked by running the main program CKZ00010. Functionality is selected by specifying the appropriate Db2 Cloning Tool command and parameters.

Required JCL varies with each command. The major factor in choosing whether COPY and RENAME will be in the CKZ00010 execution in separate steps, or in separate jobs, is whether resumption of source volume access will begin after the COPY step, or after the RENAME step.

The format of the Db2 Cloning Tool journal records can change between releases, so a journal data set created by one release of Db2 Cloning Tool should not be used by a different release of Db2 Cloning Tool. If you need to use an older release of the journal, use the JRNLUPGRADE command.

The Db2 Cloning Tool commands are listed in the following table and also explained in detail in separate topics.

| Db2 Cloning Tool Subsystem Cloning Command | Description |
|---|---|
| “BACKINFO-REFORMAT” on page 372 | Optional command that takes a backinfo data set created by the DB2GETBACKINFO command and reformats it for use by subsequent COPY commands and optionally a VOLOPTIONS command. |
| “BCSCLEAN” on page 379 | Optional command to delete target catalog entries from previous execution. |
| “COPY” on page 383 | Initiates volume copies, and in parallel, backs up the source ICF catalogs that point to data sets on the source volumes being cloned. |
| “COPY-BY-DS” on page 404 | Optional command to copy a list of data sets, as determined by RENAME masks, to target data sets with new high level qualifiers. |
| “COPYCHECK” on page 412 | Optional command to wait for completion of COPY events, or withdraw from them. |
| “DB2ALTERBSDS” on page 414 | Optional command to alter the contents of a target BSDS beyond what DB2UPDATE does. |
| “DB2FIX” on page 420 | Optional command to correct Db2 page spaces that are in restricted status. |
| “DB2GETBACKINFO” on page 424 | Optional command to issue HSM LIST COPYPOOL commands to determine the available backups from Db2 BACKUP SYSTEM and read the results to determine the source and backup volume pairs to be used. |
| “DB2LGRNXCLEAN” on page 430 | Optional command to clean out Db2 SYSLGRNX as part of removing other members. |
| “DB2RBLDBSDS” on page 432 | Optional command to rebuild a Db2 BSDS to remove other members or to make it non-data sharing. |

Table 72. Db2 Cloning Tool commands overview (continued)

| Db2 Cloning Tool Subsystem Cloning Command | Description |
|--|--|
| “DB2SCHEMA-UPDATE” on page 435 | Optional command to re-create some types of objects that cannot be processed by the Db2 CATMAINT utility with new schema values, and invokes the Db2 CATMAINT utility to change schema values for the rest of the objects. |
| “DB2SETLOG” on page 440 | Optional command to suspend or resume a Db2 subsystem. |
| “DB2SQL” on page 442 | Optional command to update the Db2 catalog. |
| “DB2START” on page 447 | Optional command to start a Db2 subsystem. |
| “DB2STOP” on page 452 | Optional command to stop a Db2 subsystem. |
| “DB2UPDATE” on page 454 | Optional command to update Db2 to reflect renamed data sets. |
| “DB2UTILXCLEAN” on page 463 | Optional command to remove utility information from the target Db2 subsystem. |
| “DB2XCFCLEAN” on page 466 | Optional command to clean out Db2 XCF structures and group members. |
| “FINDUCATS” on page 468 | Optional command to locate catalogs involved with source volume data sets. |
| “JRNLUPLGRADE” on page 470 | Optional command to upgrade a journal created by a prior release of Db2 Cloning Tool. |
| “ONLINECLIP” on page 472 | Optional command to re-label the target volume(s) when the source volume label was copied but the UCB field still points to the target volume label. This can occur when TSO FCESTABL was used. |
| “RENAME” on page 474 | Renames and catalogs data sets on target volumes. |
| “RESTORE-FROM-DUMPTAPES” on page 489 | Optional command that allows Db2 Cloning Tool to use the backups on tape from a Db2 BACKUP SYSTEM DUMP utility as the source for cloning. |
| “UCATOPTIONS” on page 496 | Optional command that will either list the user catalog pairs from the Db2 Cloning Tool journal or allow the target user catalog name(s) to be changed. |
| “VARYOFF” on page 499 | Optional command to vary target or source volumes offline. |
| “VARYON” on page 503 | Optional command to vary target or source volumes online. |
| “VOLOPTIONS” on page 507 | Optional command to use when the COPY command is run at one site and the RENAME command is run at another. |

BACKINFO-REFORMAT

This command is not required. BACKINFO-REFORMAT will take a backinfo data set created by the DB2GETBACKINFO command and reformat it for use by subsequent COPY commands and optionally a VOLOPTIONS command.

This command is used as part of the process to clone from a Db2 BACKUP SYSTEM backup. A detailed description of this cloning process can be found in [“Cloning scenarios” on page 1091](#).

The input backinfo data set consists of two types of records. The VOLMAP records identify the source volumes and their corresponding backup volumes. The UCAT records identify the source ICF catalogs and the source volumes they reside on.

The output FROM-VOLSER-DDN data set consists of a list of the backup volumes. This data set will be used in a subsequent COPY to a FROM-VOLSER-DDN keyword.

The output VOLPAIRS-DDN data set consists of source and backup volume pairs. This data set will be used in a subsequent COPY to a VOLPAIRS-DDN keyword.

The output USERCATALOGS-DDN data set consists of source ICF catalog with volser and target ICF catalog pairs. This data set will be used in a subsequent COPY to a USERCATALOGS-DDN keyword. The USERCATALOGS keyword is used to specify source and target pairs of ICF catalogs. The source ICF catalog volser will be filled in based on the UCAT records in the backinfo data set.

The output USERSGDEFS-DDN data set consists of a list of the backup volumes and the storage group names of the corresponding source volumes. This data set will be used in a subsequent COPY to a USERSGDEFS-DDN keyword. To use this data set in COPY, the COPY command must include a USERSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLEXCL(18)) keyword. This allows the FROM-USER-STORAGEGROUP keyword to be used in the COPY command, so the backup volumes can be paired to the target volumes by using the storage group names of the source and target volumes.

The output VOLOPTIONS-CMD-DDN data set consists of a generated VOLOPTIONS command. This data set will be used in a subsequent VOLOPTIONS command. The generated command will be:

```
VOLOPTIONS xxxx -  
  JOURNAL - DDN(JOURNAL)
```

Where xxxx will be LIST or TARGETUNCLIP. TARGETUNCLIP will be inserted if CLIP-IF-OFFLINE(Y) is specified and any backup volumes were clipped in this run. Otherwise, LIST will be inserted. The VOLOPTIONS-CMD-DDN keyword is intended to be used when the backup volumes are offline and have an internal volser of their corresponding source volume. BACKINFO-REFORMAT will clip the backup volumes and vary them online. After the backup volumes have been copied to the target volumes the VOLOPTIONS TARGETUNCLIP command can be used to return the backup volumes to their original state (offline with internal volser of source volume).

When CLIP-IF-OFFLINE(Y) and the backinfo VOLMAP record does not give a unique volser for a backup volume, BACKINFO-REFORMAT will generate a volser to use for the backup volume based on the masks in the VOLSER-RENAME-MASKS keyword.

BACKINFO-REFORMAT command syntax

```
BACKINFO-REFORMAT
```

Required Keywords:

```
BACKINFO-DDN( ddname )  
{ FROM-VOLSER-DDN( ddname ) | USERSGDEFS-DDN( ddname ) }  
VOLPAIRS-DDN( ddname )
```

Optional Keywords:

```
CLIP-IF-OFFLINE( N | Y )  
RESUME  
SIMULATE  
USERCATALOGS( sourcecat1 targetcat1 ...[ , sourcecatn targetcatn ] )  
USERCATALOGS-DDN( ddname )  
VOLOPTIONS-CMD-DDN( ddname )
```

Required only if CLIP-IF-OFFLINE is specified:

```
VOLSER-RENAME-MASKS( sourcemask1 backupmask1 ...[ , sourcemaskn backupmaskn ] )
```

VOLSER-RENAME-MASKS Considerations

Oldvalue Syntax

The old value filter mask is used to select the source volume volser values against which to apply the new value mask. The allowable filter characters are shown in the following table:

| <i>Table 73. Filter characters allowed for oldvalue filter masks:</i> | |
|---|---|
| Character | Description |
| * | A single asterisk represents 0 to nn characters of any value. |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| < | A lesser-than sign represents one non-numeric character, national symbols included. |
| > | A greater-than sign represents one numeric character. |

For example, Old value filter mask = SRC* would match source volume volser = SRC001.

For information about filters and ACS masks, refer to the topic [“Filtering pattern masks”](#) on page 26.

Newvalue Syntax

The new value filter mask is used to rename the source volume volser value selected by the old value filter mask. The allowable filter characters are shown in the following table:

| <i>Table 74. Filter characters allowed for newvalue filter masks</i> | |
|--|---|
| Character | Description |
| * | A single asterisk represents 0 to nn characters of any value. The single asterisk may only be used as the last character of the mask. |
| % | A percent sign represents one non-blank character. |

For example: Source volume volser = SRC001, New value filter mask = BKP*, New backup volume volser = BKP001.

For information about filters and ACS masks, refer to the topic [“Filtering pattern masks”](#) on page 26.

BACKINFO-REFORMAT command and keyword definitions

Required keywords are described first, followed by optional keywords.

Required BACKINFO-REFORMAT keywords

BACKINFO-REFORMAT

Optional command to take a backinfo data set created by the DB2GETBACKINFO command and reformat it for use by subsequent COPY commands and optionally a VOLOPTIONS command.

- Required: No
- Restrictions: None

BACKINFO-DDN(ddname)

This parameter specifies the DD name that points to a file containing the backinfo data. BACKINFO-DDN must have an LRECL of 80.

- Default: None
- Required: Yes
- Restrictions: None

FROM-VOLSER-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the from volser information will be written.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with USERSGDEFS-DDN.

USERSGDEFS-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the backup volume and source storage group information will be written.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with FROM-VOLSER-DDN.

VOLPAIRS-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the volpairs information will be written.

- Default: None
- Required: Yes
- Restrictions: None

Optional BACKINFO-REFORMAT keywords**CLIP-IF-OFFLINE(*N* | *Y*)**

This parameter specifies that Db2 Cloning Tool may clip offline backup volumes. If CLIP-IF-OFFLINE is not specified, or specified with N and a backup volume is offline, the process fails.

If CLIP-IF-OFFLINE(Y) is specified, offline backup volumes will be clipped and varied online. The offline backup volumes are expected to have an internal volser that matches their corresponding source volume volser. The clipped volser of a backup volume is determined by the value given in the backinfo data set if unique or the masks specified in the VOLSER-RENAME-MASKS keyword.

- Default: N
- Required: No
- Restrictions: None

RESUME

RESUME specifies that processing should resume for any volumes that failed to be completely processed by the previous run.

Prior to running with RESUME, the problem that caused a volume to not be completely processed should be resolved

- Default: None
- Required: No
- Restrictions: Only applies when CLIP-IF-OFFLINE(Y) is specified.

SIMULATE

SIMULATE will verify the syntax, determine the volumes to be processed, and display what action would have been taken, but will not change any volume serials with ICKDSF or vary any volumes online.

- Default: None
- Required: No
- Restrictions: Only applies if CLIP-IF-OFFLINE(Y) is specified

USERCATALOGS (*sourcecat1 targetcat1, ..., sourcecatn targetcatn*)

This parameter specifies source catalogs that data sets from source volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in. The specified source catalogs must also be in the backinfo data set in a UCAT record.

- Default: None
- Required: No
- Restrictions: Can only be specified if the backinfo data set has UCAT records.

USERCATALOGS-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the usercatalogs information will be written.

- Default: None
- Required: Required if USERCATALOGS specified.
- Restrictions: Required if USERCATALOGS specified.

VOLOPTIONS-CMD-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the VOLOPTIONS command will be written.

The VOLOPTIONS command will have the LIST option if no volumes are clipped by BACKINFO-REFORMAT and will have the TARGETUNCLIP option if volumes are clipped by BACKINFO-REFORMAT.

This parameter is intended to be used when the backup volumes are offline and are being clipped and varied online for a COPY command, after which the backup volumes will be varied offline and clipped back to their corresponding source volume volser.

- Default: None
- Required: No
- Restrictions: None

VOLSER-RENAME-MASKS (*sourcemark1 backupmask1, ... sourcemarkn backupmaskn*)

VOLSER-RENAME-MASKS are specified in 'oldvolser' 'newvolser' pairs. VOLSER-RENAME-MASKS are processed in order. The first hit of the source volser is the one that is used for the backup volser.

- Default: None
- Required: Required if CLIP-IF-OFFLINE(Y) is specified.
- Restrictions: Only applies when CLIP-IF-OFFLINE(Y) is specified.

BACKINFO-REFORMAT step JCL examples

Two BACKINFO-REFORMAT step JCL examples are included: an example that uses FROM-VOLSER-DDN to put out the backup volume information, and an example that uses USERSGDEFS-DDN to put out the backup volumes and the corresponding source storage group name information. Sample JCL can be found in the installation library SCKZJCL in member CKZBKIRF.

BACKINFO-REFORMAT Step JCL – example that uses FROM-VOLSER-DDN to put out backup volume information

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Complete BACKINFO-REFORMAT command control statement syntax is documented in the topic [“BACKINFO-REFORMAT command syntax”](#) on page 373.

The BACKINFO-REFORMAT step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'CKZ BACKINFO-REF' , CLASS=A , MSGCLASS=X
2 //S0      EXEC PGM=IDCAMS
3 //SYSPRINT DD  SYSOUT=*

```

```

//SYSIN DD *
1 DEL CKZ.WRK.VOLPAIRS
2 DEL CKZ.WRK.FRVOLSER
3 DEL CKZ.WRK.UCATS
  SET MAXCC=0
4 //S1 EXEC PGM=CKZ00010,REGION=8M
5 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
6 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
7 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
8 //BACKINFO DD DISP=SHR,DSN=CKZ.WRK.BACKINFO
9 //VOLPAIRS DD DSN=CKZ.WRK.VOLPAIRS,
  // DISP=(,CATLG),UNIT=SYSALLDA,
  // SPACE=(CYL,(1,1))
10 //FRVOLSER DD DSN=CKZ.WRK.FRVOLSER,
  // DISP=(,CATLG),UNIT=SYSALLDA,
  // SPACE=(CYL,(1,1))
11 //UCATS DD DSN=CKZ.WRK.UCATS,
  // DISP=(,CATLG),UNIT=SYSALLDA,
  // SPACE=(CYL,(1,1))
  //CKZIN DD *
    BACKINFO-REFORMAT -
8 BACKINFO-DDN(BACKINFO) -
9 VOLPAIRS-DDN(VOLPAIRS) -
10 FROM-VOLSER-DDN(FRVOLSER) -
11 USERCATALOGS-DDN(UCATS) -
12 USERCATALOGS(
  USERCAT.SRC01 USERCAT.TGT01 -
  USERCAT.SRC02 USERCAT.TGT02 -
)

```

1. Deletion of volpairs data set in anticipation of allocating new for each execution.
2. Deletion of from volser data set in anticipation of allocating new for each execution.
3. Deletion of usercatalogs data set in anticipation of allocating new for each execution.
4. Execution of Db2 Cloning Tool main program.
5. Db2 Cloning Tool SCKZLOAD library (must be authorized).
6. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
7. DD for CKZPRINT output.
8. The backinfo data set. This data set is used to pass information to the BACKINFO-REFORMAT command from the DB2GETBACKINFO command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used.
9. The volpairs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent Db2 Cloning Tool COPY command. In the sample JCL, the control statement VOLPAIRS-DDN(VOLPAIRS) specifies that a DD statement with the name VOLPAIRS is used.
10. The from volser data set that is created by BACKINFO-REFORMAT and will be used by a subsequent Db2 Cloning Tool COPY command. In the sample JCL, the control statement FROM-VOLSER-DDN(FRVOLSER) specifies that a DD statement with the name FRVOLSER is used.
11. The usercatalogs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent Db2 Cloning Tool COPY command. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.
12. The ICF user catalog pairs that will be validated and written to the USERCATALOGS-DDN. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.

BACKINFO-REFORMAT Step JCL – example that uses USERSGDEFS-DDN to put out backup volume and corresponding source storage group name information

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Complete BACKINFO-REFORMAT command control statement syntax is documented in the topic [“BACKINFO-REFORMAT command syntax”](#) on page 373.

The BACKINFO-REFORMAT step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'CKZ BACKINFO-REF' ,CLASS=A,MSGCLASS=X
      //S0 EXEC PGM=IDCAMS
      //SYSPRINT DD SYSOUT=*
      //SYSIN DD *
1     DEL CKZ.WRK.VOLPAIRS
2     DEL CKZ.WRK.USRSGDEF
3     DEL CKZ.WRK.UCATS
      SET MAXCC=0
4     //S1 EXEC PGM=CKZ00010,REGION=8M
5     //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
6     //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
7     //CKZPRINT DD SYSOUT=*
      //SYSUDUMP DD SYSOUT=*
8     //BACKINFO DD DISP=SHR,DSN=CKZ.WRK.BACKINFO
9     //VOLPAIRS DD DSN=CKZ.WRK.VOLPAIRS,
      // DISP=(,CATLG),UNIT=SYSALLDA,
      // SPACE=(CYL,(1,1))
10    //USRSGDEF DD DSN=CKZ.WRK.USRSGDEF,
      // DISP=(,CATLG),UNIT=SYSALLDA,
      // SPACE=(CYL,(1,1))
11    //UCATS DD DSN=CKZ.WRK.UCATS,
      // DISP=(,CATLG),UNIT=SYSALLDA,
      // SPACE=(CYL,(1,1))
      //CKZIN DD *
      BACKINFO-REFORMAT -
8     BACKINFO-DDN(BACKINFO) -
9     VOLPAIRS-DDN(VOLPAIRS) -
10    USRSGDEFS-DDN(USRSGDEF) -
11    USERCATALOGS-DDN(UCATS) -
12    USERCATALOGS( -
      USERCAT.SRC01 USERCAT.TGT01 -
      USERCAT.SRC02 USERCAT.TGT02 -
      )

```

1. Deletion of volpairs data set in anticipation of allocating new for each execution.
2. Deletion of USRSGDEFS data set in anticipation of allocating new for each execution.
3. Deletion of usercatalogs data set in anticipation of allocating new for each execution.
4. Execution of Db2 Cloning Tool main program.
5. Db2 Cloning Tool SCKZLOAD library (must be authorized).
6. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
7. DD for CKZPRINT output.
8. The backinfo data set. This data set is used to pass information to the BACKINFO-REFORMAT command from the DB2GETBACKINFO command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used.
9. The volpairs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent Db2 Cloning Tool COPY command. In the sample JCL, the control statement VOLPAIRS-DDN(VOLPAIRS) specifies that a DD statement with the name VOLPAIRS is used.
10. The backup volser and source storage group data set that is created by BACKINFO-REFORMAT and will be used by a subsequent Db2 Cloning Tool COPY command. In the sample JCL, the control statement USRSGDEFS-DDN(USRSGDEF) specifies that a DD statement with the name USRSGDEF is used.
11. The usercatalogs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent Db2 Cloning Tool COPY command. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.
12. The ICF user catalog pairs that will be validated and written to the USERCATALOGS-DDN. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.

BCSCLEAN

This command is not required. BCSCLEAN is intended for situations where the target catalog may be used for data sets other than those involved with a Db2 Cloning Tool process.

BCSCLEAN and its keywords support three distinct types of catalog cleanup:

- Uncatalog files in a target catalog that were renamed via the RENAME command.
- Uncatalog other files in a target catalog and on target volumes via the CLEANUP-CATALOG-ORPHANS keyword.
- Delete other files in a target catalog by data set name mask via the CLEANUP-CATALOG-DSNMASKS keyword.

For more information about CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS, refer to the keyword descriptions and the step JCL example in the topic [“BCSCLEAN Step JCL - example to clean up target user catalog entries”](#) on page 382.

If an empty target catalog is possible, rather than use BCSCLEAN, the target catalog can simply be deleted and re-defined prior to the Db2 Cloning Tool COPY step. If the redefined catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the catalog.

BCSCLEAN deletes (with no scratch) all catalog entries created in a target catalog by a previous RENAME step. BCSCLEAN is intended to delete target catalog entries created from a previous run of the Db2 Cloning Tool process that may be orphaned as a result of target volume contents being replaced.

Specifically, BCSCLEAN addresses situations where a data set used in a previous application cycle no longer exists in the current application cycle.

Additionally, for persistent data sets (application data sets that exist in every cycle), by emptying the target catalog (delete all entries from a previous run of the Db2 Cloning Tool process), the RECATALOG option of the RENAME command can be omitted, and any existing catalog entry can be treated as a true error.

Note: BCSCLEAN cannot be used after cloning by data set using the COPY-BY-DS command. To clean up the target catalog after cloning using COPY-BY-DS, use the IDCAMS DELETE command. The member CKZCBDCL in the SCKZJCL library contains sample JCL to delete all catalog entries created by the COPY-BY-DS command.

BCSCLEAN command syntax

BCSCLEAN commands syntax.

```
BCSCLEAN
```

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
CLEANUP-CATALOG-DSNMASKS( dsnmask1 , ... dsnmaskn )  
CLEANUP-CATALOG-ORPHANS  
SIMULATE
```

BCSCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

BCSCLEAN

Optional command to delete target catalog entries from previous execution.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

CLEANUP-CATALOG-DSNMASKS(*dsnmask1* , ... *dsnmaskn*)

Use this parameter to delete and uncatalog data sets that are not on a target volume, but are in a target user catalog, and as a result of the clone are no longer required. For example, there may be Db2 archive logs that were not part of the cloning process and are not needed on the target. Specify one or more data set name masks from most likely to least likely to match. The masks are checked against the list of target volumes in the journal. The first match ends the comparison. The following types of data sets are uncataloged and deleted; any other data set types are ignored:

- VSAM clusters: If the VSAM cluster name matches any individual data set name mask, the cluster is deleted and scratched. Note: Only the VSAM cluster name is used in the comparison; the VSAM data and index names are not compared.
- Non-VSAM data sets are deleted and scratched if the data set name matches any individual mask.
- GDGs: Only GDS files within a GDG are considered for deletion. Each GDS is a unique non-VSAM file and is individually considered for deletion (as above). A GDS is deleted and scratched if its data set name matches any individual mask. The GDG base file itself is not uncataloged or deleted; the user must manually delete the GDG base.

In some cases, deleting migrated files may cause HSM (or equivalent software) to attempt to recall the file, before delete processing can proceed.

An optional audit log file can be specified to capture a list of files deleted during CLEANUP-CATALOG-DSNMASKS processing. To specify the log file, include a DD named LOGFILE, defined as SYSOUT=*, or as a data set with DSORG=PS and LRECL=80, or as a PDS member with the same DCB characteristics). All files that are processed for deletion are logged, including failures for objects not found or other errors.

- Default: None
- Required: No
- Restrictions: None

CLEANUP-CATALOG-ORPHANS

Use this parameter to uncatalog data sets that are no longer on the target volume after the cloning process, but are still in the target user catalog. For example, there may be non-VSAM data sets or

VSAM clusters that existed on the target volume before the clone, but were not part of the cloning process. The data sets are checked against the list of target volumes in the journal to determine whether the data set should be uncataloged. If the data set is on multiple volumes, all must be target volumes as defined in the journal in order to be deleted. However, candidate volumes are ignored. The following types of data sets are uncataloged; any other data set types are ignored:

- VSAM clusters: All volumes for the VSAM data and index components must be target volumes as defined in the journal file. Otherwise, the file is bypassed and logged with first failing volume and associated data or index component.
- Non-VSAM files: All volumes must be target volumes as defined in the journal file. Otherwise, the file is bypassed and logged with first failing volume.
- GDGs: GDS files within a GDG are considered for deletion. Each GDS is a unique non-VSAM file and is individually considered for deletion (as above). The GDG base file itself is not uncataloged or deleted; the user must manually delete the GDG base.

Note: For each data set, CLEANUP-CATALOG-ORPHANS processing occurs before CLEANUP-CATALOG-DSNMASKS processing. This impacts data sets that could potentially match both keywords (those that are on a target volume and also match a supplied *dsnmask* value, as follows:

- Because a match on any target volume means the data set was deleted by the cloning process, a data set that could match both keywords is processed only by CLEANUP-CATALOG-ORPHANS, and only uncataloged (no delete from the volume is attempted).
- No CLEANUP-CATALOG-DSNMASKS processing is required for such data sets. This can increase efficiency, especially when many data set name masks are supplied.

An optional audit log file can be specified to capture a list of files deleted during CLEANUP-CATALOG-ORPHANS processing. To specify the log file, include a DD named LOGFILE, defined as SYSOUT=*, or as a data set with DSORG=PS and LRECL=80, or as a PDS member with the same DCB characteristics. All files that are processed for deletion are logged, including failures for objects not found or other errors.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Include this parameter to generate a list of data sets and information about whether they will be uncataloged or deleted during a subsequent non-SIMULATE run. No updates of any kind are performed. Output is generated in CKZPRINT with other BCSCLEAN output.

It is possible that the number of files identified in a SIMULATE run as candidates for deletion differs from the number actually removed by CLEANUP-CATALOG-ORPHANS or CLEANUP-CATALOG-DSNMASKS during the normal (non-SIMULATE) run. In a normal run, standard BCSCLEAN processing occurs first, which might remove some catalog entries that CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS processing then will not encounter. In a SIMULATE run, the two separate BCSCLEAN processes each review the exact same user catalog records, because no deletes or updates are performed.

- Default: None
- Required: No
- Restrictions: None

BCSCLEAN step JCL examples

This topic contains two examples of BCSCLEAN step JCL.

BCSCLEAN Step JCL example

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Sample JCL can be found in the installation library SCKZJCL in member CKZBCLN.

Note that the 'BCSRECS' data set specified in the JCL contains the list of catalog entries that were put there by the prior RENAME and are to be deleted.

The BCSCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'CKZ BCSCLEAN', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //BCSRECS DD DSN=CKZ.WRK.BCSRECS, DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
9 //CKZIN DD *
10 BCSCLEAN
11 JOURNAL - DDN(JOURNAL)
12 /*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for BCSRECS. The BCSRECS data set contains the names of data sets cataloged in a previous RENAME step. The names contained in this data set determine the catalog entries to be deleted by BCSCLEAN.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

BCSCLEAN Step JCL - example to clean up target user catalog entries

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

Note that the 'BCSRECS' data set specified in the JCL contains the list of catalog entries that were put there by the prior RENAME and are to be deleted.

The BCSCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'CKZ BCSCLEAN', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //BCSRECS DD DSN=CKZ.WRK.BCSRECS, DISP=SHR
6 //CKZPRINT DD SYSOUT=*
```

```

//SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL,DISP=OLD
//CKZINI DD *
BCSCLEAN -
7 CLEANUP-CATALOG-ORPHANS -
8 CLEANUP-CATALOG-DSNMASKS(DB1T.ARCHLOG*.**) -
6 JOURNAL-DDN(JOURNAL)
//*

```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for BCSRECS. The BCSRECS data set contains the names of data sets cataloged in a previous RENAME step. The names contained in this data set determine the catalog entries to be deleted by BCSCLEAN.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. CLEANUP-CATALOG-ORPHANS parameter that specifies to uncatalog data sets that would no longer be on the target volume after the cloning process, but would still be in the target user catalog.
8. CLEANUP-CATALOG-DSNMASKS parameter that specifies to delete and uncatalog data sets that are not on a target volume, but are in a target user catalog. In this case, each DB1T.ARCHLOG**.**) data set will be checked against the list of target volumes in the journal to determine whether the data set is to be deleted and uncataloged.

COPY

This command is required. COPY invokes volume copies via FlashCopy or SnapShot if the DATA-MOVER(PGM(ADRDSSU)) is specified, or invokes volume copies via TimeFinder/Clone Mainframe Snap Facility's volume level support if the DATA-MOVER(PGM(EMCSNAP)) is specified, or assumes copies have been created by the user if DATA-MOVER(PGM(NONE)) is specified. In all cases, COPY captures catalog data pertaining to source volume data sets.

Important: The ICF catalog backup can be postponed until after COPY by using the USERCATALOGS-NOBACKUP keyword. However, if you choose to postpone the backup, the source ICF catalog information being backed up needs to be in synchronization with the contents of the copied volumes. If USERCATALOGS-NOBACKUP is used (so the source ICF catalogs are backed up after COPY by UCATOPTIONS BACKUP), the user must ensure that the source ICF catalogs are logically at the same point in time as when the volumes were copied.

About source and target catalog names

It is valid for source and target catalog names to be the same. If source and target catalog names are the same, the target catalog will be populated. See the [RECATALOG](#) option of the [“RENAME”](#) on page 474 and the [“BCSCLEAN”](#) on page 379 command.

For example, when:

- Source volumes contain data sets named A1... and B1...:
- Alias A1 points to UCATA1 and alias B1 points to UCATB1
- Alias A2 points to UCATA2 and alias B2 points to UCATB2

- Rename masks (see “RENAME” on page 474): A1.** A2.** B1.** B2.** USERCATALOGS(UCATA1 UCATA2 UCATB1 UCATB2)

Then:

1. Any source volume data sets matching A1.** are renamed to A2.**.
2. A2.** data sets are cataloged to UCATA2 because source data sets A1.** were found to be cataloged in UCATA1.
3. Any source volume data sets matching B1.** are renamed to B2.**.
4. B2.** data sets are cataloged to UCATB2 because source data sets B1.** were found to be cataloged in UCATB1.

Note: Catalog data must be captured that reflects the status of the source volume data sets at the time of the copies. Some data needed to catalog renamed data sets, such as catalog PATH entries and GDG base records, exist in the catalog only. Db2 Cloning Tool does not attempt to ascertain involved catalogs automatically, due to the time this would take and the consequent delay of source volume access.

Volume pairing process and criteria

The pairing of source and target volumes consists of two stages:

1. The first stage pairs any target volumes that have target catalogs on them if the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter is set. In this case, for each target volume that has a target catalog, a source volume with a source catalog on it is located. All target volumes with target catalogs on them must be paired at this stage; otherwise, the COPY command completes with an error. If there are no target volumes with target catalogs on them, the first stage is not required.
2. In the second stage of pairing, for each source volume, a target volume is located.

In addition, there are essential pairing requirements for the ADRDSSU utility and for the EMCSNAP utility. If the following requirements are not fulfilled, volumes cannot be paired:

- If a fast replication method is specified for ADRDSSU (FASTREP PREF or FASTREP REQ), then the source and the target volumes are checked for the possibility of pairing using fast replication.
- If a slow copy method is specified for ADRDSSU (FASTREP NONE), or fast replication requirements cannot be met for ADRDSSU (FASTREP PREF), then the source and target volumes are checked to ensure they are the same device type.
- If EMCSNAP is used, the following conditions are checked:
 - Both devices are EMC (Symmetrix) devices.
 - Both devices have the required microcode level.
 - Both devices have similar device serial numbers.
 - Both devices are the same device type (for example, both devices are 3390s).

For ADRDSSU, pairing of volumes that have catalogs on them depends on the value that is specified in the FASTREP parameter, as follows:

- If FASTREP(REQ) is specified, volumes can only be paired using the fast replication method.
- If FASTREP(NONE) is specified, volumes can only be paired using the slow copy (normal) method.
- If FASTREP(PREF) is specified, for all target volumes:
 - Db2 Cloning Tool first attempts to pair using the fast replication method.
 - If a target volume is not paired, then a pairing is selected that will use the slow copy method.

First stage

All target volumes that have catalogs on them are sorted by size, in descending order. All source volumes that have catalogs on them are grouped by volume size. For each target volume, Db2 Cloning Tool attempts to match a source volume that is less than or equal to the target volume size. Source volumes that were paired with the target by using one of the USERCATALOGS keywords are checked first. Then Db2

Cloning Tool continues until all of the source volumes are checked. The next steps depend on the following conditions:

- If a slow copy method was specified, or a fast replication method using FASTREP(REQ) or EMCSNAP was specified, the volumes cannot be paired, and the volume pairing completes with an error.
- If fast replication method FASTREP(PREF) was specified, then Db2 Cloning Tool proceeds to the next target volume.
- Finally, if fast replication method FASTREP(PREF) was specified, and the target volumes that have catalogs on them cannot be paired using fast replication requirements, Db2 Cloning Tool attempts to pair the volumes using the slow copy method.

At the end of the first stage, all of the target volumes with target catalogs on them should be paired. If not, the COPY completes with an error.

Second stage

All non-paired target volumes are grouped by volume size. All source volumes are sorted by volume size in ascending order. For each source volume, Db2 Cloning Tool attempts to find a matching target volume with a size greater or equal to the source volume size. Db2 Cloning Tool proceeds to check each target volume until all of the target volumes are checked. The next steps depend on the following conditions:

- If a slow copy method or fast replication method with FASTREP(REQ) or EMCSNAP was specified, the volume pairing completes with an error.
- If a fast replication method was specified with FASTREP(PREF), then Db2 Cloning Tool proceeds to the next source volume and repeats the process.
- Finally, if fast replication method FASTREP(PREF) was specified, and the source volumes cannot be paired using fast replication requirements, Db2 Cloning Tool attempts to pair the volumes using the slow copy method.

At the end of the second stage, all of the source volumes should be paired. If not, the COPY completes with an error.

How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified

If you want to ensure that source volumes are paired with specific target volumes, use the KEEP-VOLUMES-SEQUENCE keyword. When this keyword is specified, Db2 Cloning Tool assigns sequence numbers to the volume specifications in the FROM and TO parameters and maps the source to the target according to those numbers.

When the COPY command is used with the FROM-STORAGEGROUP, FROM-USER-STORAGEGROUP, FROM-VOLSER, or FROM-VOLSER-DDN parameters, then for each source volume that was defined by one of these parameters, a sequence number is defined that consists of two parts: *type* and *position*.

- *type* of a source volume sequence number is a keyword that describes the parameter that is used for gathering of this volume, and is one of the following:
 - STOGRP if the volume is gathered from the FROM-STORAGEGROUP parameter.
 - USRSGR if the volume is gathered from the FROM-USER-STORAGEGROUP parameter.
 - VOLUME if the volume is gathered from the FROM-VOLSER or FROM-VOLSER-DDN parameters.
- *position* of the sequence number is the position that this volume is in for the corresponding parameter, starting with 0.

For example, for the following parameter:

```
FROM-VOLSER(VOLSR1, VOLSR2, VOLP*)
```

The sequence numbers are defined as:

- VOLSR1 is VOLUME 0.
- VOLSR2 is VOLUME 1.

- All volumes that meet the VOLP* mask (such as VOLP01, VOLP02, and so on) are VOLUME 2.

In the same manner, when the COPY command uses the TO-STORAGEGROUP, TO-USER-STORAGEGROUP, TO-VOLSER or TO-VOLSER-DDN parameters, for each target volume that is defined by one of these parameters, a sequence number is defined that consists of two parts: *type* and *position*.

- *type* of a source volume sequence number is a keyword that describes the parameter that is used for gathering of this volume, and is one of the following:
 - STOGRP if the volume is gathered from the TO-STORAGEGROUP parameter.
 - USRSGR if the volume is gathered from the TO-USER-STORAGEGROUP parameter.
 - VOLUME if the volume is gathered from the TO-VOLSER or TO-VOLSER-DDN parameters.
- *position* of the sequence number is the position that this volume is in for the corresponding parameter, starting with 0.

For example, for the following parameter:

```
TO-VOLSER(VOLTG1, VOLTG2, VOLM*)
```

The sequence numbers are defined as:

- VOLTG1 is VOLUME 0.
- VOLTG2 is VOLUME 1.
- All volumes that meet the VOLM* mask (such as VOLM01, VOLM02, and so on) are VOLUME 2.

FlashCopy for backups

When Db2 Cloning Tool runs the COPY command, it uses the FlashCopy default of 'background COPY' from the source to target volume after the logical completion occurs.

If you want to copy and rename the target volume data sets to be used as input to a backup, FlashCopy's 'NO background COPY' (FCNOCOPY) should be used. If you take a point-in-time copy using FlashCopy, and the target volume is only needed for a short time, such as for input to a backup, copying tracks that haven't changed would be a waste of resources compared to just using the pointer to the corresponding source volume data set. This concept is called 'NO background COPY'. The 'before image' of tracks that change on the source volume must in fact be created on the target. But, assuming that a DFSMSdss or FDR DUMP of the target volume is started just after the FlashCopy initiation is complete, most likely not many source volume tracks will change in the time it takes the backup to finish.

When the backup is finished however, this NO background COPY process should be stopped to prevent changed data from continually being copied to the target volume.

If you use DFSMSdss to back up the FlashCopy target, a DFSMSdss DUMP parameter, FlashCopy Withdraw (FCWITHDRAW), can be used on the DFSMSdss DUMP to tell DSS to withdraw the FlashCopy relationship when the backup is complete.

If you use FDR as your dump tool, Db2 Cloning Tool provides the COPYCHECK WITHDRAW command to withdraw the FlashCopy relationship. For more information, see the topic "[COPYCHECK](#)" on page 412.

Note: If either the source or target volumes are extensively updated during the backup, this option should not be used because excessive overhead will occur when copying changed tracks to the target before allowing the update to occur.

COPY command syntax

```
COPY
```

Required Keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }  
USERCATALOGS( sourcecat1 [ ( sourcevolser1 ) ] targetcat1 [ ( targetvolser1 ) ] ...
```

```
[ , sourcecatn [ ( sourcevolsern ) ] targetcatn [ ( targetvolsern ) ] ] ) |
USERCATALOGS-DDN( ddname ) |
USERCATALOGS-NOBACKUP( sourcecat1 [ ( sourcevolser1 ) ] targetcat1 [ ( targetvolser1 ) ] ...
[ , sourcecatn [ ( sourcevolsern ) ] targetcatn [ ( targetvolsern ) ] ] ) |
USERCATALOGS-BACKUPFIRST( sourcecat1 targetcat1 ...[ , sourcecatn
targetcatn ] ) |
NOUSERCATALOGS
```

Required only if DATA-MOVER(PGM(ADRDSU)) is specified (FlashCopy or SnapShot) or DATA-MOVER(PGM(EMCSNAP)) is specified (EMC TimeFinder/Clone volume snap):

```
{ [ FROM-STORAGEGROUP( storgrp1 | storgrpmask1 ...[ , storgrpn | storgrpmaskn ] ) ]
[ FROM-USER-STORAGEGROUP( storgrp1 | storgrpmask1 ...
[ , storgrpn | storgrpmaskn ] ) ]
[ FROM-VOLSER( volser1 | volmask1 ...[ , volsern | volmaskn ] )
| FROM-VOLSER-DDN( ddname ) ] }
{ [ TO-STORAGEGROUP( storgrp1 | storgrpmask1 ...[ , storgrpn | storgrpmaskn ] ) ]
[ TO-USER-STORAGEGROUP( storgrp1 | storgrpmask1 ...
[ , storgrpn | storgrpmaskn ] ) ]
[ TO-VOLSER( volser1 | volmask1 ...[ , volsern | volmaskn ] )
| TO-VOLSER-DDN( ddname ) ] }
```

Required only if DATA-MOVER(PGM(NONE)) is specified:

```
{ VOLPAIRS( sourcevolser1 targetvolser1 ...[ , sourcevolsern targetvolsern ] ) |
VOLPAIRS-DDN( ddname ) |
VOLPAIRSDEVN( sourcevolser1 targetvolser1 devn1 ... [ , sourcevolsern
targetvolsern devnn ] ) | VOLPAIRSDEVN-DDN( ddname ) |
VOLPAIRSDEVN-NOCLIP ( sourcevolser1 targetvolser1 devn1
, sourcevolsern targetvolsern devnn [ , ... ] ) |
VOLPAIRSDEVN-NOCLIP-DDN ( ddname ) }
```

Required only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP is specified:

```
USERSGDEFS-DDN( ddname )
USERSGDEFS-OFFSETS( INCLXCL(nn), SGNAME(nn), VOLSER(nn) )
```

Required only if USERCATALOGS, USERCATALOGS-NOBACKUP, USERCATALOGS-BACKUPFIRST, or USERCATALOGS-DDN is specified:

```
CATWORK-DSN( mask )
```

Optional Keywords:

```
BASE-JOURNAL-DDN ( ddname )
CATWORK-ATTR( UNIT( SYSALLDA ) SPACE( 10 10 ) CYLINDERS )
DATA-MOVER( [ PGM( ADRDSU | EMCSNAP | NONE ) ]
[ , BACKGROUNDCOPY( NO | YES ) ]
[ , CHECKONLINEPATHSTATUS( NO | YES ) ]
[ , CHECKVTOC ]
[ , CONSISTENT( NO | YES ) ]
[ , DIFFERENTIAL( NO | YES ) ]
[ , DSSPAMP( ABEND=nnn [ , AMSGCNT=nnnn ] | SDUMP=nnn [ , SMSGCNT=nnnn ] ) ]
[ , COPYCMDLIMIT( nnn | 24 ) ]
[ , FASTREP( PREF | REQ | NONE ) ]
[ , FCNOCOPY ]
[ , FCSETGTOK ]
[ , FCTOPPRCPRIARY [ ( PRESMIRREQ | PRESMIRPREF | PRESMIRNONE ) ]
[ , FCTOXRCPRIARY ]
[ , INCREMENTAL( NO | YES ) ]
[ , NOCONCURRENT ]
[ , MAXIMUM-SUBTASKS( number1, number2 ) ] )
EXCLUDE-FROM-VOLSER( volser1 | volmask1 ...[ volsern | volmaskn ] )
EXCLUDE-TO-VOLSER( volser1 | volmask1 ...[ volsern | volmaskn ] )
KEEP-VOLUMES-SEQUENCE ( Y | N )
SIMULATE
SOURCESONLINE( Y | N )
TARGETSONLINE( Y | N )
TARGETSUONLINE( Y | N )
TARGET-UCATS-ON-TARGET-VOLUMES( Y | N )
TARGET-VOLS-SHOULD-BE-EMPTY( Y | N )
```

COPY command and keyword definitions

Required keywords are described first, followed by optional keywords.

COPY

The COPY command initiates volume copies and in parallel, backs up the source ICF catalogs that point to data sets on the source volumes being cloned.

- Required: Yes
- Restrictions: None

CATWORK-DSN (*mask*)

Specifies a mask used to derive data sets names for catalog backup data sets dynamically allocated during the COPY step.

The mask must include an asterisk (*) as one qualifier. Db2 Cloning Tool will create data sets by substituting two eight-byte qualifiers in place of the provided asterisk. Hence, because 17 bytes (8+the dot+8) of the name will be generated, the user is responsible for the resolved names not exceeding 44 characters (e.g., CATWORK-DSN(CKZ.CATWORK.*) will cause data sets to be created such as: CKZ.CATWORK.UCATBKUP.BKP00001

The asterisk in the mask does not need to be the lowest level qualifier. (e.g., CATWORK(CKZ.CWORK*.DATA)

- Default: None
- Required: Required if USERCATALOGS, USERCATALOGS-NOBACKUP, USERCATALOGS-BACKUPFIRST, or USERCATALOGS-DDN is specified.
- Restrictions: None
- Short form: CWDSN

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

Optional syntax is: JRNL-DSN or JRNL-DDN

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: JRNL-DSN JRNL-DDN

USERCATALOGS (*sourcecat1* [(*sourcevolser1*)] *targetcat1* [(*targetvolser1*)] ... [, *sourcecatn* [(*sourcevolsern*)] *targetcatn* [(*targetvolsern*)]])

This parameter specifies the source ICF catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in.

ICF catalog names are specified in pairs of source and target. For each renamed data set, the USERCATALOGS list is searched for the catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

The source ICF catalogs specified must include all ICF catalogs that any data set being copied and renamed could be cataloged in.

Target ICF catalogs that are used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step, unless the TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword of the COPY command is specified. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

If you specify the *sourcevolser* with the source ICF catalog name, the source catalog will be backed up from its copy on the corresponding target volume. The *sourcevolser* that is specified is the source volume where the source ICF catalog resides. The volume where the source ICF catalog resides must be included as one of the source volumes being copied. The VOLSER of the source ICF catalog is specified because when Db2 Cloning Tool does the volume pairing, there is no guarantee the source volume will be paired to the same target volume for every cloning. Db2 Cloning Tool uses the volume pairing information to determine the target volume VOLSER that corresponds with the specified source volume VOLSER.

The *targetvolser* value can be specified with the target ICF catalog name to indicate the target volume that corresponds with the location of the copied source ICF catalog. This value is only used if TARGET-UCATS-ON-TARGET-VOLUMES(Y) has been specified. The *targetvolser* value is used with some cloning scenarios when the volume pairing is not done via a COPY command to allow the proper renaming of the cloned ICF catalogs.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NOUSERCATALOGS, USERCATALOGS-BACKUPFIRST, USERCATALOGS-NOBACKUP, and USERCATALOGS-DDN.
- Short form: UCATS

USERCATALOGS-NOBACKUP (*sourcecat1* [(*sourcevolser1*)] *targetcat1* [(*targetvolser1*)] ... [, *sourcecatn* [(*sourcevolsern*)] *targetcatn* [(*targetvolsern*)]])

This parameter specifies source catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in.

ICF catalog names are specified in pairs of source and target. For each renamed data set, the USERCATALOGS list is searched for the ICF catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

The source ICF catalogs specified must include all ICF catalogs that any data set being copied and renamed could be cataloged in.

Target ICF catalogs that are used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step, unless the TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword of the COPY command is specified. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

If you specify the *sourcevolser* with the source ICF catalog name, the source catalog will be backed up from its copy on the corresponding target volume. The *sourcevolser* that is specified is the source volume where the source ICF catalog resides. The volume where the source ICF catalog resides must be included as one of the source volumes being copied. The VOLSER of the source ICF catalog is specified because when Db2 Cloning Tool does the volume pairing, there is no guarantee the source volume will be paired to the same target volume for every cloning. Db2 Cloning Tool uses the volume pairing information to determine the target volume VOLSER that corresponds with the specified source volume VOLSER.

The *targetvolser* value can be specified with the target ICF catalog name to indicate the target volume that corresponds with the location of the copied source ICF catalog. This value is only used if TARGET-UCATS-ON-TARGET-VOLUMES(Y) has been specified. The *targetvolser* value is used with some cloning scenarios when the volume pairing is not done via a COPY command to allow the proper renaming of the cloned ICF catalogs.

The source catalogs will not be backed up by COPY. The source catalogs will be backed up when UCATOPTIONS BACKUP is run. The backup of the source catalogs with UCATOPTIONS BACKUP must happen prior to running the RENAME command. If VOLSER is specified for the source catalog, UCATOPTIONS BACKUP will make its backup from the copies of the source catalogs on the target volumes.

Note: The ICF catalog backup can be postponed until after COPY by using the USERCATALOGS-NOBACKUP keyword. However, if you choose to postpone the backup, the source ICF catalog information being backed up needs to be in synchronization with the contents of the copied volumes. If USERCATALOGS-NOBACKUP is used (so the source ICF catalogs are backed up after COPY by UCATOPTIONS BACKUP), the user must ensure that the source ICF catalogs are logically at the same point in time as when the volumes were copied.

If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN are used with USERCATALOGS-NOBACKUP, VOLOPTIONS TARGETOFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NOUSERCATALOGS, USERCATALOGS, USERCATALOGS-DDN, and USERCATALOGS-BACKUPFIRST.
- Short form: UCATSNB

USERCATALOGS-BACKUPFIRST (*sourcecat1 targetcat1 ... [, sourcecatn targetcatn]*)

This parameter specifies source ICF catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in.

ICF catalog names are specified in pairs of source and target. For each renamed data set, the USERCATALOGS list is searched for the catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

The source ICF catalogs specified must include all ICF catalogs that any data set being copied and renamed could be cataloged in.

Target ICF catalogs that are used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step, unless the TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword of the COPY command is specified. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

The source ICF catalogs will be backed up before the target volumes are processed by the COPY command.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NOUSERCATALOGS, USERCATALOGS, USERCATALOGS-DDN, and USERCATALOGS-NOBACKUP.
- Short form: UCATSBF

USERCATALOGS-DDN(*ddname*)

This parameter specifies the DD name which points to a file containing user catalog pairs. The pairs are the same format as in the USERCATALOGS keyword.

USERCATALOGS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with USERCATALOGS, NOUSERCATALOGS, USERCATALOGS-NOBACKUP, and USERCATALOGS-BACKUPFIRST.
- Short form: UCATSDDN

NOUSERCATALOGS

Specifies that the COPY will not include the backing up of any ICF catalogs. NOUSERCATALOGS is intended to be used for the following cases:

- When creating a copy of the source volumes where the target volumes are strictly for backup purposes (there is no intent of renaming the data sets on the target volumes).
- When performing a volume cloning using interim volumes. Refer to [“Cloning scenarios” on page 1091](#) for more information about this type of cloning.

RENAME is not possible when NOUSERCATALOGS is used.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with USERCATALOGS, USERCATALOGS-DDN, USERCATALOGS-NOBACKUP and USERCATALOGS-BACKUPFIRST.
- Short form: NOUCATS

FROM-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[, *storgrpn* | *storgrpmaskn*])

Specifies the input volumes to be copied from one or more SMS storage group definitions or storage groups matching a mask. All volumes from the storage groups specified will be copied, except any volumes excluded via the EXCLUDE-FROM-VOLSER parameter.

An input (source) storage group may only be specified or referred to once per execution. Db2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, same volume size, and the target volume size is equal to or greater than the source volume size.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385](#).

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSSU)) and DATA-MOVER(PGM(EMCSNAP))
- Short form: FRS

FROM-USER-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[*storgrpn* | *storgrpmaskn*]) or FROMUSERSTORAGEGROUP

Specifies the input volumes to be copied from one or more user storage group definitions or storage groups matching a mask. All online volumes from the storage groups specified will be copied, except any volumes excluded via the EXCLUDE-FROM-VOLSER parameter.

An input (source) storage group may only be specified or referred to once per execution. Db2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, same volume size, and the target volume size is equal to or greater than the source volume size.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385](#). The storage group definitions that are used will be read from the DD statement defined by the USERSGDEFS-DDN keyword.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSSU)) and DATA-MOVER(PGM(EMCSNAP))
- Short form: FRUS

FROM-VOLSER (*volser1* | *volmask1* ...[, *volsern* | *volmaskn*])

Optional syntax: FROMVOLSER

Specifies the input volumes to be copied, either by discrete volume serial numbers or volume serial masks (e.g., TSO*). See the EXCLUDE-FROM-VOLSER keyword to exclude VOLSERS from a list.

Db2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, same volume size, and the target volume size is equal to or greater than the source volume size.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385.](#)

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRSSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified.
- Short form: FRV

FROM-VOLSER-DDN(*ddname*)

Optional syntax: FROMVOLSERDDN

This parameter specifies the DD name that points to a file containing FROM VOLSER volumes. The volumes are the same format as in the FROM-VOLSER keyword.

FROM-VOLSER-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385.](#)

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRSSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified.
- Short form: FRVDDN

TO-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[, *storgrpn* | *storgrpmaskn*])

Optional syntax: TOSTORAGEGROUP

Specifies that output volumes needed to pair with input volumes are to be selected from one or more SMS storage groups or storage groups matching a mask. All volumes from the storage groups specified are target candidates, except any volumes excluded via the EXCLUDE-TO-VOLSER parameter.

Db2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, same volume size, and the target volume size is equal to or greater than the source volume size.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385.](#)

If FROM-USER-STORAGEGROUP is used, but FROM-STORAGEGROUP or TO-USER-STORAGEGROUP are not used, Db2 Cloning Tool will attempt to match volumes using the same positioned storage group names that are listed in TO-STORAGEGROUP. If KEEP-VOLUMES-SEQUENCE(Y) is also

specified, then each FROM-USER-STORAGEGROUP volume will be paired with the TO-STORAGEGROUP volume that has the same positioned storage group name. If that is not possible, then a pairing error occurs.

If FROM-STORAGEGROUP is used, Db2 Cloning Tool will attempt to match volumes using the same positioned storage group names listed in TO-STORAGEGROUP. If a hardware difference or other issue results in a FROM-STORAGEGROUP volume that does not match its TO-STORAGEGROUP counterpart, the TO-STORAGEGROUP volume is skipped. Db2 Cloning Tool then attempts to match the FROM-STORAGEGROUP volume with the next TO-STORAGEGROUP volume in the list. If KEEP-VOLUMES-SEQUENCE(Y) is also specified, then each FROM-STORAGEGROUP volume is paired with the TO-STORAGEGROUP volume that has the same positioned storage group name. If that is not possible, then a pairing error occurs.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSU)) and DATA-MOVER(PGM(EMCSNAP))
- Short form: TOS

TO-USER-STORAGEGROUP (*storgrp1* | *storgmask1* ...[, *storgprn* | *storgmaskn*])

Optional syntax: TOUSERSTORAGEGROUP

Specifies that output volumes needed to pair with input volumes are to be selected from one or more user storage groups or storage groups matching a mask. All online volumes from the storage groups specified are target candidates, except any volumes excluded via the EXCLUDE-TO-VOLSER parameter.

Db2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, same volume size, and the target volume size is equal to or greater than the source volume size.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385.](#)

If FROM-STORAGEGROUP is used but FROM-USER-STORAGEGROUP or TO-STORAGEGROUP are not used, Db2 Cloning Tool will attempt to match volumes using the same positioned storage group names that are listed in TO-STORAGEGROUP. If KEEP-VOLUMES-SEQUENCE(Y) is also specified, then each FROM-USER-STORAGEGROUP volume is paired with the TO-STORAGEGROUP volume that has the same positioned storage group name. If that is not possible, then a pairing error occurs.

If FROM-USER-STORAGEGROUP is used, Db2 Cloning Tool attempts to match volumes using the same positioned storage group names listed in TO-USER-STORAGEGROUP. If the FROM-USER-STORAGEGROUP volume does not match its TO-USER-STORAGEGROUP counterpart because of hardware differences or other issues, the TO-USER-STORAGEGROUP volume is skipped. Db2 Cloning Tool then attempts to match the FROM-USER-STORAGEGROUP volume with the next TO-USER-STORAGEGROUP volume in the list. If KEEP-VOLUMES-SEQUENCE(Y) is also specified, then each FROM-USER-STORAGEGROUP volume is paired with the TO-USER-STORAGEGROUP volume that has the same positioned storage group name. If that is not possible, then a pairing error occurs.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSU)) and DATA-MOVER(PGM(EMCSNAP))
- Short form: TOUS

TO-VOLSER (*volser1* | *volmask1* ...[, *volsern* | *volmaskn*])

Optional syntax: TOVOLSER

Specifies target volumes to be paired with input volumes.

Db2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, same volume size, and the target volume size is equal to or greater than the source volume size.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385.](#)

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRSSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified.
- Short form: TOV

TO-VOLSER-DDN(*ddname*)

Optional syntax: TOVOLSERDDN

This parameter specifies the DD name that points to a file containing to volser volumes. The volumes are the same format as in the TO-VOLSER keyword.

TO-VOLSER-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

If KEEP-VOLUMES-SEQUENCE(Y) is specified, make sure to specify the desired source and target volume pairs so that they have the same sequence number. The assignment of sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified” on page 385.](#)

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRSSU)) or DATA-MOVER(PGM(EMCSNAP)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified. Mutually exclusive with TO-VOLSER.
- Short form: TOVDDN

USERSGDEFS-DDN (*ddname*)

Specifies a DD name which points to a file containing the user storage group definitions that will be used by the FROM-USER-STORAGEGROUP and TO-USER-STORAGEGROUP keywords. The USERSGDEFS-OFFSET keyword is used to specify the offsets of the fields within the records.

The file must have an LRECL of 80. A comment record is denoted by a blank or * in column 1 or a /* in columns 1 and 2.

- Default: None
- Required: Required only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP are specified.
- Restrictions: Used only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP are specified.

USERSGDEFS-OFFSETS (*INCLXCL(nn)*, *SGNAME(nn)*, *VOLSER(nn)*)

Specifies the offsets in the record that the fields will be found. These fields are used for the user storage group definitions that will be used by the FROM-USER-STORAGEGROUP and TO-USER-STORAGEGROUP keywords.

The offsets specified can be from 1 to 72 inclusive, with 1 being the first character position in the record.

INCLEXCL specifies the offset where the 1-character include/exclude indicator can be found. A value of blank or I indicates an include definition. A value of E or X indicates an exclude definition.

SGNAME specifies the offset where the 8-character storage group name can be found.

VOLSER specifies the offset where the 6-character volser or volser mask can be found.

For example: Using `USERSGDEFS-OFFSETS(VOLSER(1), INCLEXCL(8), SGNAME(10))` and `USERSGDEFS-DDN` file containing the records:

```
SRC00*  SGSRC01
SRC006 E SGSRC01
TGT01*  I SGTGT01
TGT02*  I SGTGT01
TGT015  X SGTGT01
TGT025  X SGTGT01
```

`FROM-USER-STORAGEGROUPS(SGSRC01)` would resolve to all online DASD volumes that match the mask `SRC00*` except for `SRC006` which would be excluded.

`TO-USER-STORAGEGROUPS(SGTGT01)` would resolve to all online DASD volumes that match the masks `TGT01*` and `TGT02*` except that `TGT015` and `TGT025` would be excluded.

- Default: None
- Required: Required only if `FROM-USER-STORAGEGROUP` or `TO-USER-STORAGEGROUP` are specified.
- Restrictions: Used only if `FROM-USER-STORAGEGROUP` or `TO-USER-STORAGEGROUP` are specified.

VOLPAIRS (*sourcevolser1 targetvolser1 ... [, sourcevolsern targetvolsern]*)

Optional syntax:

- `VOLPAIRS-DDN(ddname) |`
- `VOLPAIRSDEVN (sourcevolser1 targetvolser1 devn1 ... [, sourcevolsern targetvolsern devnn])`
- `VOLPAIRSDEVN-DDN(ddname) |`
- `VOLPAIRSDEVN-NOCLIP (sourcevolser1 targetvolser1 devn1 ... [, sourcevolsern targetvolsern devnn])`
- `VOLPAIRSDEVN-NOCLIP-DDN (ddname)`

VOLPAIRS specifies volume pairs where the target volumes have been created by the user before the Db2 Cloning Tool `COPY` command is executed. Target volumes must have the desired internal VOLSER (not the VOLSER of the source volume) and be online. Discrete VOLSERs only, no masking allowed. Short form is `VP`.

VOLPAIRS-DDN specifies a DD name which points to a file containing the volume pairs. The pairs are the same format as in the `VOLPAIRS` keyword. `VOLPAIRS-DDN` must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

VOLPAIRSDEVN specifies volume pairs with the device number of the target volume where target volumes have been created by the user before the Db2 Cloning Tool `copy` command is executed. Target volumes must be offline and the internal VOLSER must match the corresponding source VOLSER specified. Db2 Cloning Tool will re-label the specified device (`devn`) to the corresponding target VOLSER and vary the volume online. Short form is `VPD`.

VOLPAIRSDEVN-DDN specifies a DD name which points to a file containing the volume pairs. The pairs are the same format as in the `VOLPAIRSDEVN` keyword. `VOLPAIRSDEVN-DDN` must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

VOLPAIRSDEVN-NOCLIP specifies volume pairs with the device number of the target volume where target volumes have been created by the user before the Db2 Cloning Tool `COPY` command is executed. Target volumes must be offline and the internal VOLSER must match the corresponding

source VOLSER specified. COPY will not re-label the specified device (devn) or vary the volume online. The VOLOPTIONS TARGETOFFLINECLIP command can be used to re-label the specified target devices and vary the volumes online. The re-label and vary online of the target volumes with VOLOPTIONS TARGETOFFLINECLIP must happen prior to running the RENAME command. If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN are used with USERCATALOGS-NOBACKUP, VOLOPTIONS TARGETOFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

VOLPAIRSDEVN-NOCLIP-DDN specifies a ddname which points to a file containing the volume pairs. The pairs are the same format as in the VOLPAIRSDEVN-NOCLIP keyword. VOLPAIRSDEVN-NOCLIP-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

- Default: None
- Required: One of the VOLPAIRS keywords is required only if DATA-MOVER(PGM(NONE)) is specified.
- Restrictions: Valid only with DATA-MOVER(PGM(NONE)).
- Short form: FRV

BASE-JOURNAL-DDN (*ddname*)

This parameter supplies the DD name of a Db2 Cloning Tool base journal file that (via JCL) points to a journal data set. The base journal data set is created by the primary COPY step and will be used by the RENAME step. Use this keyword when the cloning scenario has multiple COPY steps and information must be passed between COPY steps.

- Default: None
- Required: No
- Restrictions: None
- Short form: BASE-JRNL-DDN

CATWORK-ATTR (*catalog backup allocation attributes*)

Specifies allocation attributes used when catalog backup data sets are dynamically allocated. Unless unusual attributes are desired for a specific Db2 Cloning Tool COPY, these attributes can be globally specified in the HLQ?.SCKZPARM. CKZINI member, :COPY_OPTIONS section, CATWORK-ATTR token. Allocation attributes are specified in TSO allocate syntax (e.g., UNIT(SYSDA) SPACE(1 1) TRACKS, etc.).

The attributes that can be specified include:

- DATACLAS(data class name)
- MGMTCLAS(management class name)
- SPACE(quantity increment)
- STORCLAS(storage class name)
- TRACKS/CYLINDERS UNIT(unit)
- VOLUME(serial)

If an initial attempt running COPY fails because a catalog backup data set exceeds extents, increase the allocation and run again. Once successful, examine the space actually used and decrease if desired. To accommodate a future increase in the size of catalogs, leave the allocation with room to spare.

- Default: UNIT(SYALLDA) SPACE(10 10) CYLINDERS
- Required: No.
- Restrictions: None
- Short form: CWATTR

DATA-MOVER ([PGM(ADRDSSU | EMCSNAP | NONE)]

For **ADR**DSSU:

```
[ , CHECKVTOC ]  
[ , CONSISTENT( NO | YES ) ]  
[ , COPYCMDLIMIT( nnn | 24 ) ]  
[ , DSSPARM( ABEND=nnn [ , AMSGCNT=nnnn ] | SDUMP=nnn [ , SMSGCNT=nnnn ] ) ]  
[ , FASTREP( PREF | REQ | NONE ) ]  
[ , FCNOCOPY ]  
[ , FCSETGTOK ]  
[ , FCTOPPRCPPRIMARY [ (PRESMIRREQ | PRESMIRPREF | PRESMIRNONE ) ]  
[ , FCTOXRCPPRIMARY ]  
[ , INCREMENTAL( NO | YES ) ]  
[ , NOCONCURRENT ] )
```

For **EMCSNAP**:

```
[ , BACKGROUNDCOPY( NO | YES ) ]  
[ , CHECKONLINEPATHSTATUS( NO | YES ) ]  
[ , CONSISTENT( NO | YES ) ]  
[ , DIFFERENTIAL( NO | YES ) ]  
[ , MAXIMUM-SUBTASKS( number1, number2 ) ] )
```

In all cases, specifies the program to be used to initiate copies and copy options.

ADRDSSU specifies that COPY is to initiate FlashCopy or SnapShot 'under the covers' via execution of DSS. It is the default and therefore the PGM(ADR DSSU) keyword can be omitted. Db2 Cloning Tool invokes DSS 'under the covers' to initiate volume copies with the COPY FULL option. Users should be acquainted with the DSS rules governing copy full operations (IBM publication Advanced Copy Services, Combining Storage Control Copy Operations).

For a COPY FULL operation, DSS determines by the device types of the volumes to be copied what copy mechanism is to be used. Db2 Cloning Tool pairs volumes, so that FlashCopy or SnapShot should be used, and confirms at the time of the COPY step that conflicting relationships do not exist. However, because of the small window between Db2 Cloning Tool volume pairing and relationship validation, DSS errors and associated messages may need to be examined.

Db2 Cloning Tool COPY command supplies the ADMINISTRATOR operand when invoking DFSMSdss. To avoid WTORs, ADR DSSU ADMINISTRATOR is used to gain permission to overlay the target volume VTOCIX and/or VVDS during the COPY process.

Because the ADMINISTRATOR operand is generated, the user ID running COPY must have READ access to FACILITY class profile STGADMIN.ADR.STGADMIN.COPY.

EMCSNAP specifies that COPY is to initiate EMC SNAP 'under the covers' via EMC TimeFinder/Clone Mainframe Snap Facility's volume level support. Db2 Cloning Tool invokes EMC TimeFinder/Clone 'under the covers' to initiate volume copies using SNAP VOLUME commands. Users should be acquainted with the EMC rules governing SNAP VOLUME operations (EMC publication TimeFinder/Clone Mainframe SNAP Facility Product Guide).

Db2 Cloning Tool pairs volumes, so that EMC TimeFinder/Clone volume snap should be used, and confirms at the time of the COPY step that conflicting relationships do not exist. However, because of the small window between Db2 Cloning Tool volume pairing and relationship validation, EMC TimeFinder/Clone volume snap errors and associated messages may need to be examined.

NONE specifies that no DATA-MOVER is to be invoked by COPY. NONE infers that volume copies have been created by the user prior to the execution of Db2 Cloning Tool. When NONE is specified, COPY still captures necessary catalog information, and if VOLPAIRSDEVN or VOLPAIRSDEVN-DDN is specified, clips and varies target volumes online.

The following parameters are for ADRDSSU:

CHECKVTOC (DSS parameter): Specifies that a VTOC analysis of the source volume be performed during copy processing.

CONSISTENT (Db2 Cloning Tool parameter): Indicates to use a FlashCopy Consistency Group (YES) or not (NO). The default is NO. This does not apply when FASTREP(NONE or PREF) is also specified. To

establish a Consistency Group, a FCCGFREEZE parameter is added to each COPY statement passed to ADRDSSU. This will cause I/O activity to the FlashCopy source volumes to be held (frozen). Once all the COPY statements have been processed by ADRDSSU, a CGCREATE command is passed to ADRDSSU. This will cause I/O activity to be resumed on the "frozen" FlashCopy source volumes. The number of volumes being copied will affect the amount of time the I/O activity to the source volumes will be held. If this time exceeds the smallest Consistency Group timer value defined for a LSS, the target volumes will not be consistent.

COPYCMDLIMIT(nnn | 24) (Db2 Cloning Tool parameter): Specifies the maximum COPY FULL commands built by Db2 Cloning Tool for each DSS execution. 24 is the default if omitted. If the number of volumes to be copied exceeds the COPYCMDLIMIT, Db2 Cloning Tool will invoke DSS as many times as necessary. Adjusting this value may affect the performance of ESS copy initiations.

DSSPARM (DSS parameter): This parameter can be used to pass execution parameters to ADRDSSU. This parameter is intended for diagnostic purposes and can be used to gather more information when ADRDSSU has problems. The parameter values should be: ABEND=nnn [, AMSGCNT=nnnn] or SDUMP=nnn [, SMSGCNT=nnnn]. These parameters are described in the DFSMS Storage Administration Reference manual in the section "How to Control DFSMSdss through PARM Information in the EXEC Statement".

FASTREP (DSS parameter): Indicates if fast replication is preferred (PREF), required (REQ), or not required (NONE). Db2 Cloning Tool will set up the source/target pairs for a fast replication if PREF or REQ is specified. Db2 Cloning Tool will allow a 'normal' copy if NONE is specified. If the level of ADRDSSU indicates it supports this keyword, the keyword will be passed to ADRDSSU.

FCNOCOPY: (DSS parameter) Indicates that no background copy should be done for the volume pair. This applies to ESS devices only.

Note: When using FCNOCOPY, you must terminate the FlashCopy source and target relationship at the completion of the backup. Either use the DFSMSdss DUMP parameter FCWITHDRAW, or use COPYCHECK WITHDRAW. Following the withdraw, some of the tracks on the volume may contain data from the source volume, while other tracks may contain residual data that was on the target volume before the copy. This situation can cause problems when trying to access the target volume if the VTOC locations of the source and target volumes were different before the copy.

FCSETGTOK (DSS parameter): Indicates that a FlashCopy target volume can also be a space efficient volume. This does not apply when FASTREP(NONE) is also specified. The physical background copy option is not permitted for space efficient FlashCopy. Because of this, the FCNOCOPY keyword must also be specified with the FCSETGTOK keyword.

FCTOPPRCPRIARY [(PRESMIRREQ | PRESMIRPREF | PRESMIRNONE)] (DSS parameter): Indicates that a FlashCopy target volume can also be a PPRC primary volume. This applies to ESS devices only. This does not apply when FASTREP(NONE) is also specified.

IBM Remote Pair FlashCopy (also known as Preserve Mirror) can be specified by including one of the optional keywords. Preserve Mirror mirrors the FlashCopy command that is issued at the local site to the remote site. This allows FlashCopy operations to occur to PPRC primary volumes without affecting the PPRC duplex state. IBM Remote Pair FlashCopy must be installed in the storage controller along with the corresponding software support in z/OS. In addition, both the source and target volumes being PPRC primary volumes and in the same storage controller and their corresponding PPRC secondary volumes being in the same storage controller.

Specify one of the following use this functionality:

- PRESMIRREQ (short form: PMR): Require the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation will not be completed.
- PRESMIRPREF (short form: PMP): Prefer the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation is still performed.
- PRESMIRNONE (short form: PMN): Do not use Preserve Mirror.

When any of these options is specified, they will be used as part of the source to target volume pairing criteria. The specified option will also be passed to ADRDSSU as part of the generated copy

commands. For additional information about Preserve Mirror, refer to the documentation for your version of z/OS.

FCTOXRCPPRIMARY (DSS parameter): Indicates that a FlashCopy target volume can also be an XRC primary volume. This keyword is applicable only for DATA-MOVER PGM(ADRDSSU). This does not apply when FASTREP(NONE) is also specified.

INCREMENTAL (Db2 Cloning Tool parameter): Indicates that a full volume Incremental FlashCopy relationship is to be established (YES) or not (NO). The default is NO. This does not apply when FASTREP(NONE or PREF) is also specified. Volumes will be paired using an existing Incremental FlashCopy relationship if possible. INCREMENTAL(YES) cannot be used with FCNOCOPY.

NOCONCURRENT (DSS parameter): Specifies that the CONCURRENT option will not be supplied to ADRDSSU. This will prevent ADRDSSU from using Concurrent Copy when doing the volume copies.

The following parameters are for EMCSNAP:

BACKGROUNDCOPY: Indicates if background copy should be done (YES), or not (NO), for the volume pair. The default is YES.

CHECKONLINEPATHSTATUS: Indicates if a check that paths from other CPUs to the target devices are offline before performing a volume snap (YES) or not (NO). The default is NO.

CONSISTENT: Indicates to use Enginuity Consistency Assist (ECA) for consistent SNAP VOLUME operations (YES) or not (NO). The default is NO.

DIFFERENTIAL: Indicates to use the Enginuity Differential Snap feature for SNAP VOLUME operations (YES) or not (NO). The default is NO.

MAXIMUM-SUBTASKS (*number1*, *number2*): Sets an absolute maximum number of subtasks that can be attached and used. SNAP automatically limits the number of subtasks based on the requests specified and the low and high region available. SNAP never exceeds the limits specified in this parameter. *number1* is the limit of the number of individual requests that can be processed simultaneously. The minimum value you can specify is two (2). The maximum value you can specify is 9999. *number2* is the limit of the number of individual activities that can be performed within a single request, typically as the result of wildcarding. The minimum value you can specify is two (2). The maximum value you can specify is 9999. If not specified, the EMC system defaults are used.

- Default: For PGM, the default is ADRDSSU. For COPYCMDLIMIT, the default is 24. For FASTREP, the default is REQ. For INCREMENTAL, the default is NO. For BACKGROUNDCOPY, the default is YES. For CHECKONLINEPATHSTATUS, the default is NO. For CONSISTENT, the default is NO. For DIFFERENTIAL, the default is NO.
- Required: No
- Restrictions: INCREMENTAL(YES) is mutually exclusive with FCNOCOPY.
- Short form(s): DM, CHECK, CCL, FR, FCNC, FCTOPPRCP, NOCC

EXCLUDE-FROM-VOLSER (*volser1* | *volmask1* ...[*volsern* | *volmaskn*])

Optional syntax: EXCLUDEFROMVOLSER

Specifies volumes or volumes matching a mask, to be excluded from being specified for either the FROM-VOLSER parameter, or the FROM-STORAGEGROUP parameter.

- Default: None
- Required: No
- Restrictions: Not valid with VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN.
- Short form: EXCFRV

EXCLUDE-TO-VOLSER (*volser1* | *volmask1* ...[*volsern* | *volmaskn*])

Optional syntax: EXCLUDETOVOLSER

Specifies volumes or volumes matching a mask, to be excluded (not selected as targets) from being specified for either the TO-VOLSER parameter or the TO-STORAGEGROUP parameter.

If the 'eliminated' target volumes cause there to be more source volumes than targets, the COPY will fail.

If the 'eliminated' target volumes still leave at least as many target volumes as source volumes, the pairing will continue as usual.

- Default: None
- Required: No
- Restrictions: Not valid with VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN.
- Short form: EXCTOV

KEEP-VOLUMES-SEQUENCE(Y | N)

Specifies whether source volumes are to be paired with specific target volumes. During the pairing process, Db2 Cloning Tool checks the volume sequence numbers of source and target volumes. The assignment of volume sequence numbers is described in detail in [“How Db2 Cloning Tool pairs volumes when the KEEP-VOLUMES-SEQUENCE keyword is specified”](#) on page 385. If Y or YES is specified, then target and source volumes will be paired only if their sequence numbers are the same. If N or NO is specified, or no KEEP-VOLUMES-SEQUENCE parameter is specified, then the sequence numbers of source and target volumes are not used for their pairing. In this case, source and target volumes that have different sequence numbers can be paired if other required conditions are true.

- Default: N
- Required: No
- Restrictions: Mutually exclusive with VOLPAIR, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, and VOLPAIRSDEVN-NOCLIP-DDN
- Short form: KVS

SIMULATE

For DATAMOVER(PGM(ADRDSSU)), SIMULATE specifies that the COPY step is to perform all functions except actual initiation of copies. This option is recommended for new Db2 Cloning Tool setups or when modifications to the setup are made.

SIMULATE will verify syntax, match source to target volumes, display DSS COPY FULL commands (but not initiate the copies), and back up user catalogs. User catalogs are backed up to enable the SIMULATE option of the RENAME step.

Note that in the process of pairing volumes, the logic is exercised to assure that the same ESS subsystem, LSS (FlashCopy V1), and volume size requirements permit all source volumes to be paired with targets. If discrete volumes are specified, and one-for-one pairing is desired, the simulated COPY FULL commands will verify whether pairing requirements were satisfied for a one-for-one pairing.

For DATAMOVER(PGM(NONE)), SIMULATE specifies that the COPY step checks syntax, and backs up the source ICF Catalogs for input to RENAME SIMULATE.

- Default: None.
- Required: No
- Restrictions: None.
- Short form: SIM

SOURCESONLINE(Y | N)

Db2 Cloning Tool COPY command will expect to locate the source volumes online. If N is specified, Db2 Cloning Tool COPY will not expect to find all the source volumes online.

Note: If N is specified and the source volumes are not online, the source user catalogs must still be available to the Db2 Cloning Tool COPY command.

- Default: Y
- Required: No.

- Restrictions: Valid only with VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN
- Short form: SRCON

TARGETSONLINE(Y | N)

If Y is specified, Db2 Cloning Tool COPY will expect to locate the target volumes online. If N is specified, Db2 Cloning Tool COPY will not expect to find all the target volumes online.

- Default: Y
- Required: No
- Restrictions: Valid only with VOLPAIRS or VOLPAIRS-DDN.
- Short form: TGTON

TARGETSUONLINE(Y | N)

If Y is specified, Db2 Cloning Tool COPY will expect to locate the target user catalogs. It will check that the security product will allow ALTER access to the target user catalogs.

If N is specified, Db2 Cloning Tool COPY will not check for the existence / availability of the target user catalogs. Db2 Cloning Tool RENAME will check that the target catalogs are available and will do the RACF checks.

- Default: Y
- Required: No.
- Short form: TGTUON

TARGET-UCATS-ON-TARGET-VOLUMES(Y | N)

Specifies that there are target ICF catalogs that reside on the target volumes, and you want the target ICF catalogs to remain on the target volumes after the RENAME step. If Y is specified, when the RENAME step completes, the target ICF catalogs will reside on target volumes.

- Default: N
- Required: No
- Restrictions: None
- Short form: TUOTV

TARGET-VOLS-SHOULD-BE-EMPTY(Y | N)

Performs a check during the volume pairing process to ensure the target volumes are empty before issuing FlashCopy or SnapShot. In the event a subsequent RENAME fails and the COPY must be rerun, Db2 Cloning Tool will not clean off the target volumes if 'Y' was specified for this parameter. Either initialize the target volumes or change this keyword to 'N'.

- Default: N
- Required: No
- Restrictions: Not valid with VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN.
- Short form: TVE

Offline Volumes (with DATA-MOVER(PGM(ADRDSU)) or PGM(EMCSNAP))

Source volumes:

- Discrete VOLSER specification: Must be found online, or COPY terminates with a return code 8.
- VOLSER mask specification: At least one online volume must be found that matches the mask, or COPY terminates with a return code 8.
- Storage group specification: At least one volume associated with the storage group must be online, or COPY terminates with a return code 8. A warning, return code 4, is generated if at least one volume is

not found. Note that Db2 Cloning Tool cannot tell the difference between a non-existent volume and an offline volume.

Target volumes (no matter how they are specified) :

- If enough online volumes can be found to pair with source volumes, COPY will proceed.
- If target volumes are not found, but not needed, COPY will not complain.
- If fewer target volumes are found than needed to pair with source volumes, COPY will terminate with a return code 8.
- Because in some cases Db2 Cloning Tool cannot distinguish between a non-existent volume and an offline volume, messages refer to these volumes as 'not found'.

COPY step JCL example

Two COPY step JCL examples are included: an example that uses ADRDSSU to do the volume copies, and an example that uses EMC TimeFinder/Clone volume snaps to do the volume copies. Sample JCL can be found in the installation library SCKZJCL in member CKZCOPY. Both ADRDSSU and EMC TimeFinder/Clone examples are contained in the member.

COPY Step JCL – example that uses ADRDSSU to do the volume copies

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The COPY step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //????????? JOB , 'CKZ COPY', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 1 DEL CKZ.JRNL
6 2 DEL CKZ.WRK.UCATBKUP.*
7 SET MAXCC=0
8 //S1 EXEC PGM=CKZ00010, REGION=8M
9 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
10 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
11 //CKZPRINT DD SYSOUT=*
12 //SYSUDUMP DD SYSOUT=*
13 7 //JOURNAL DD DSN=CKZ.JRNL, RECOG=KS, KEYLEN=64, KEYOFF=0,
14 // DISP=(,CATLG), UNIT=SYSALLDA,
15 // LRECL=600, SPACE=(CYL,(10,10))
16 //CKZIN DD *
17 COPY
18 DATA-MOVER(
19 )
20 FROM-VOLSER(VSRC02)
21 TO-VOLSER(VTGT02)
22 8 USERCATALOGS(
23 USERCAT.SRC01 USERCAT.TGT01
24 USERCAT.SRC02 USERCAT.TGT02
25 )
26 2 CATWORK-DSN(CKZ.WRK.* )
27 7 JOURNAL-DDN(JOURNAL)
//*
```

1. Deletion of journal data set in anticipation of allocating new for each execution. Because this data set is used to pass information from one Db2 Cloning Tool step to another, do not delete the journal data set in any steps except the COPY step.
2. Deletion of catalog backup work data sets in anticipation of allocating new for each execution. The Db2 Cloning Tool COPY step backs up each catalog specified by the USERCATALOGS control statement. Output data sets for each catalog backup are dynamically allocated using the CATWORK-DSN(mask) to derive names. Allocation attributes are specified in the CKZINI member of SCKZPARM: COPY_OPTIONS section, CATWORK-ATTR token. Delete these data sets only in the COPY step.
3. Execution of Db2 Cloning Tool main program.

4. Db2 Cloning Tool SCKZLOAD library must be authorized.
5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
6. DD for Db2 Cloning Tool output.
7. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. Each Db2 Cloning Tool 'application' must have a unique journal data set.

Because logging varies based on errors and or warnings discovered in the process, the data set size is difficult to predict. Therefore, test an application using the suggested CYL 10,10 allocation and increase if necessary. After a successful execution, the size may be reduced after examining the total allocation. Allow for changes to the application setup. Number of volumes, number of data sets and number of warnings will affect the required size.

As noted in the IDCAMS step, be sure this data set is not deleted before all Db2 Cloning Tool steps are complete.

In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD statement with the name JOURNAL is used rather than a dynamically allocated data set name.

8. During the COPY process, Db2 Cloning Tool produces two work files for each catalog pair specified in USERCATALOGS:
 - BKPnnnnn is the backup of the source catalogs.
 - SRTnnnnn is used as the output from the RENAME sort of the BCS records selected by the RENAME-MASKS.

COPY Step JCL – example that uses EMC TimeFinder/Clone volume snaps to do the volume copies

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The COPY step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //????????? JOB , 'CKZ COPY' ,CLASS=A,MSGCLASS=X
2 //S1 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 DEL CKZ.JRNL
6 DEL CKZ.WRK.UCATBKUP.*
7 SET MAXCC=0
8 //S1 EXEC PGM=CKZ00010,REGION=8M
9 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
10 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
11 //CKZPRINT DD SYSOUT=*
12 //SYSUDUMP DD SYSOUT=*
13 //JOURNAL DD DSN=CKZ.JRNL,RECOG=KS,KEYLEN=64,KEYOFF=0,
14 // DISP=(,CATLG),UNIT=SYSALLDA,
15 // LRECL=600,SPACE=(CYL,(10,10))
16 //CKZIN DD *
17 COPY -
18 DATA-MOVER( -
19 PGM(EMCSNAP) -
20 ) -
21 FROM-VOLSER(VSRC02) -
22 TO-VOLSER(VTGT02) -
23 USERCATALOGS( -
24 USERCAT.SRC01 USERCAT.TGT01 -
25 USERCAT.SRC02 USERCAT.TGT02 -
26 ) -
27 CATWORK-DSN(CKZ.WRK.* ) -
28 JOURNAL-DDN(JOURNAL)
29 /**

```

1. Deletion of journal data set in anticipation of allocating new for each execution. Because this data set is used to pass information from one Db2 Cloning Tool step to another, do not delete the journal data set in any steps except the COPY step.
2. Deletion of catalog backup work data sets in anticipation of allocating new for each execution. The Db2 Cloning Tool COPY step backs up each catalog specified by the USERCATALOGS control statement. Output data sets for each catalog backup are dynamically allocated using the CATWORK-DSN(mask) to derive names. Allocation attributes are specified in the CKZINI member of SCKZPARM:COPY_OPTIONS section, CATWORK-ATTR token. Delete these data sets only in the COPY step.
3. Execution of Db2 Cloning Tool main program.
4. Db2 Cloning Tool SCKZLOAD library must be authorized.
5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
6. DD for Db2 Cloning Tool output.
7. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. Each Db2 Cloning Tool 'application' must have a unique journal data set.

Because logging varies based on errors and or warnings discovered in the process, the data set size is difficult to predict. Therefore, test an application using the suggested CYL 10,10 allocation and increase if necessary. After a successful execution, the size may be reduced after examining the total allocation. Allow for changes to the application setup. Number of volumes, number of data sets and number of warnings will affect the required size.

As noted in the IDCAMS step, be sure this data set is not deleted before all Db2 Cloning Tool steps are complete.

In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD statement with the name JOURNAL is used rather than a dynamically allocated data set name.

8. Requests that the volumes copies be done using EMC TimeFinder/Clone Mainframe Snap Facility's volume level support.
9. During the COPY process, Db2 Cloning Tool produces two work files for each catalog pair specified in USERCATALOGS:
 - BKPnnnnn is the backup of the source catalogs.
 - SRTnnnnn is used as the output from the RENAME sort of the BCS records selected by the RENAME-MASKS.

COPY-BY-DS

This command is not required. This command is used only for Db2 subsystem cloning by data set. The COPY-BY-DS command copies a list of data sets, as determined by RENAME masks, to target data sets with new high level qualifiers. This feature allows you to copy all or a subset of data sets for a Db2 subsystem to a smaller number of volumes for a new Db2 subsystem.

Volume copy and rename can be faster than cloning a subsystem by data set, but there might be cases where it is not feasible or desirable to clone by volume. Cloning a subsystem by data set allows you to copy all or a subset of data sets for a Db2 subsystem to another subsystem. The target Db2 subsystem can have a smaller number of volumes than the source subsystem.

The COPY-BY-DS command copies a list of data sets, as determined by RENAME masks, to target data sets with new high level qualifiers.

Planning for subsystem cloning by data set

The process for Db2 subsystem cloning by data set is different from the process for cloning by volume. The COPY-BY-DS command replaces the COPY and RENAME commands that are used when cloning

subsystems by volume. Review this topic before using the COPY-BY-DS command to clone a subsystem by data set.

Requirements

The COPY-BY-DS command supports IBM fast replication via ADRDSSU; FASTREP(PREF) is the default. COPY-BY-DS applies new data set names and catalogs on new volumes per SMS rules.

The following are required for cloning by data set:

- Both the source and the target Db2 subsystems must be stopped for the COPY-BY-DS step.
- Target data sets must be SMS managed to assign space and disk volumes.
- Db2 storage groups must all be SMS managed. For more information, see the topic [“DB2SQL” on page 442](#).
- You cannot clone ICF catalogs or other types of catalogs by using the COPY-BY-DS command.

Tips for cloning subsystems by data set

The following tips provide guidance for cloning subsystems by data set with the COPY-BY-DS command:

- Migrated data sets that are included by matching a RENAME-MASKS (source) mask name will cause the underlying COPY to fail (i.e., ADRDSSU COPY). If you want to clone data sets that have been migrated, you must recall them in a separate step before the cloning. To determine if the data sets that you want to clone have been migrated, first run the COPY-BY-DS command with the SIMULATE option. This option generates a report that lists the data sets to be cloned, along with any data sets that are migrated and therefore ineligible to be cloned. Examine the report and recall any migrated data sets that you want to clone. Any migrated data sets that you do not want to clone can be excluded from the copy by using either the EXCLUDE-SRCNAME-MASKS keyword with a list of the migrated data sets or masks that will match only the migrated data sets, or by using the MIGRATED-DSN(SKIP) keyword to exclude (skip) all migrated data sets.
- When running repetitive clones, use the REPLACE-UNCONDITIONAL keyword to allow the COPY-BY-DS command to reuse existing allocated files where possible. As an alternative, omit the REPLACE-UNCONDITIONAL keyword, but add an IDCAMS DELETE command to delete the target data sets before the COPY-BY-DS step. Failure to use one of these two techniques (or a functional equivalent) will result in a return code of 8 on repetitive clones.
- Sometimes errors may occur during the underlying COPY itself, such as ENQ and other resource contention errors. COPY-BY-DS produces a return code of 8 in these cases. Look for message CKZ06531E to find these errors, or if none of those messages are found, look for ADRxxxE messages. These errors should be resolved before trying to rerun COPY-BY-DS.
- A journal file is required and used in later steps for Db2 conditioning. For COPY-BY-DS, there is no RESUME or RESTART capability. Therefore, before rerunning COPY-BY-DS after a failure, you must delete the journal and start with a new one.
- BCSCLEAN cannot be used after cloning by data set using the COPY-BY-DS command. To clean up the target catalog after cloning using COPY-BY-DS, use the IDCAMS DELETE command. The member CKZCBDCL in the SCKZJCL library contains sample JCL to delete all catalog entries created by the COPY-BY-DS command.

For more information about the steps and requirements for cloning by data set, see the cloning scenario topic [“Db2 subsystem cloning using data set copy” on page 1103](#).

RENAME considerations for COPY-BY-DS

When renaming data sets as part of subsystem cloning by data set, you should consider the following information when specifying RENAME-MASKS.

- RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.

- Name lengths: Whether changing a qualifier to a longer new name qualifier, or using the + feature to add qualifiers, be careful that new names do not exceed 44 characters.
- Rename collisions: Be sure rename masks cannot cause two or more old names to rename to the same new name.
- If VSAM data sets are used, then masks must define both cluster components for the source and the target, or both data components for the source and the target. For example, for Db2 data sets, masks can be either *srchlq.DSNDBC.** tghlq.DSNDBC.***, or *srchlq.DSNDBD.** tghlq.DSNDBD.***.
- Only the Db2 logs and BSDSs can be renamed beyond the *hlq*. All other Db2 files expect the following:
 - Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBC.DSNDB01.**
 - Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBC.DSNDB06.**
 - Db2 databases – Db2 expects a specific naming standard: *vcat.DSNDBC.dbname.psname.y0001.Annn*

Oldname syntax

Use the old name filter mask to select the data sets to apply the RENAME new name mask against. The allowable filter characters are shown in the following table:

| <i>Table 75. Filter characters allowed for old name filter masks</i> | |
|--|--|
| Character | Description |
| * | A single asterisk represents exactly one DSN qualifier of any value. Example: *. Or .* or *.*. * combined with valid DSN characters or % means 0 to <i>nn</i> characters of any value. |
| ** | A double asterisk represents 0 to <i>nn</i> DSN qualifiers of any value. For example: **, or .**, or .** ** cannot appear with any other characters within a qualifier. Three or more adjacent * are not allowed within a qualifier. ** can be used more than once in an 'old name' mask. Example: **.abcd.** or abc.**.defg.** |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| < | A less-than sign represents one non-numeric character, national symbols included. |
| > | A greater-than sign represents one numeric character. |

Newname syntax

Use the new name mask to rename the data sets selected by the old name filter mask. The allowable filter characters are shown in the following table:

| <i>Table 76. Filter characters allowed for new name filter masks</i> | |
|--|--|
| Character | Description |
| * | A single asterisk represents exactly one DSN qualifier. * may not be used for a partial qualifier in a 'new name' mask. Example: <i>aaa.*bb.**</i> would not be valid. |

Table 76. Filter characters allowed for new name filter masks (continued)

| Character | Description |
|-----------|---|
| ** | A double asterisk represents 0 to <i>nn</i> DSN qualifiers of any value. For example: **. or **. or **. ** cannot appear with any other characters within a qualifier. Three or more adjacent * are not allowed within a qualifier. ** can be used more than once in an 'old name' mask. Example: **.abcd.** or abc.**.defg.** |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| +cccccc | A plus sign followed by 1 to 8 characters means 'insert this new qualifier'. |
| - | A minus sign means 'remove this qualifier from the new name'. |

For example:

Original DSN/Mask: CKZI . LAB9 . DEM01 . SDS . ORIG

Rename To DSN/Mask: CKZI . +TEST . XX%% . - . **

New Name: CKZI . TEST . XXB9 . SDS . ORIG

COPY-BY-DS command syntax

COPY-BY-DS

Required Keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
{ RENAME-MASKS ( mask pairs ) | RENAME-MASKS-DDN( ddname ) }
```

Optional Keywords:

```
DATA-MOVER( [ PGM( ADDRSSU ) ]
  [ , DSNS-PER-COPY ( nnn | 255 ) ]
  [ , DSS-COPY-COMMANDS ( nnn | 5 ) ]
  [ , DSSPARM( ABEND=nnn [ , AMSGCNT=nnnn ] | SDUMP=nnn [ , SMSGCNT=nnnn ] ) ]
  [ , FASTREP ( PREF | REQ | NONE ) ]
  [ , FCTOPPRCPRIARY [ (PRESMIRREQ | PRESMIRPREF | PRESMIRNONE ) ]
  [ , FCTOXRCPRIARY ]
  EXCLUDE-SRCNAME-MASKS ( sourcemask1, sourcemask2, ... sourcemaskn )
  | EXCLUDE-SRCNAME-MASKS-DDN ( ddname )
  MIGRATED-DSN( ERROR | SKIP )
  MGMTCLAS( mgmtclass ) | NULLMGMTCLAS
  OFFLINE
  REPLACE-UNCONDITIONAL
  SIMULATE
  STORCLAS( storclass ) | NULLSTORCLAS
```

COPY-BY-DS command and keyword definitions

Required keywords are described first, followed by optional keywords.

COPY-BY-DS

The COPY-BY-DS command copies a list of data sets, as determined by RENAME data set name masks, to target data sets with new high level qualifiers. The COPY-BY-DS command replaces the COPY and RENAME sequence of commands used with traditional subsystem cloning.

- Required: No
- Restrictions: None

**JOURNAL-DSN (*data set name*)
or JOURNAL-DDN (*ddname*)**

Optional syntax is: JRNL-DSN or JRNL-DDN

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple Db2 Cloning Tool setups are used for different volume groups or subsystems, DO NOT use the same journal data set. Each Db2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps. Therefore, as noted in the JCL comments, it must be cataloged in the COPY-BY-DS step when first created, and referenced as OLD in subsequent steps.

A journal file is required, and used in later steps for Db2 conditioning. For COPY-BY-DS, there is no restart or rerun capability. After a failure during COPY-BY-DS, you must delete the journal and start with a new journal before rerunning.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: JRNL-DSN or JRNL-DDN

RENAME-MASKS (*mask pairs*)

RENAME-MASKS are specified in 'oldname' 'newname' pairs. RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name. RENAME-MASKS can be specified so that the target data set names will be exactly the same as the source data set names.

All data sets can be renamed beyond the high level qualifier, except for the following:

- Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBC.DSNDB01.**
- Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBC.DSNDB06.**
- Db2 databases – Db2 expects a specific naming standard:
vcat.DSNDBC.dbname.pname.y0001.Annn

Db2 bootstrap data sets and Db2 logs can be renamed beyond the high level qualifier.

An example of correctly specified masks follows:

```
RENAME-MASKS (DB2APROD.** DB2CPROD.** –
              DB2BPROD.** DB2DPROD.**)
```

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with RENAME-MASKS-DDN.

RENAME-MASKS-DDN (*ddname*)

This parameter specifies a DD name that points to a file containing rename mask pairs. The pairs are the same format as in the RENAME-MASKS keyword. RENAME-MASKS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with RENAME-MASKS.

DATA-MOVER ([PGM(ADRDSSU)]
[, DSNS-PER-COPY (nnn | 255)]
[, DSS-COPY-COMMANDS (nnn | 5)]
[, DSSPARM(ABEND=nnn [, AMSGCNT=nnnn] | SDUMP=nnn [, SMSGCNT=nnnn])]
[, FASTREP(PREF | REQ | NONE)]
[, FCTOPPRCPPRIMARY [(PRESMIRREQ | PRESMIRPREF | PRESMIRNONE)]
[, FCTOXRCPPRIMARY]

Specifies the program to be used to initiate copies and copy options.

ADRDSSU specifies that COPY is to initiate FlashCopy or SnapShot 'under the covers' via execution of DSS. It is the default and therefore the PGM(ADRDSSU) keyword can be omitted.

DSNS-PER-COPY specifies the number of data sets to include in each COPY statement. The maximum and the default value is 255. This value interacts directly with DSS-COPY-COMMANDS; the recommended maximum for these two values multiplied together is 1400 or less. Values that are too high may cause resource shortage errors, particularly with the TIOT.

DSS-COPY-COMMANDS specifies the number of COPY commands to pass in each unique call to ADRDSSU. The default value of 5 is recommended. The maximum value is 10. This value interacts directly with DSNS-PER-COPY; the recommended maximum for these two values multiplied together is 1400 or less. Values that are too high may cause resource shortage errors, particularly with the TIOT.

DSSPARM (DSS parameter): This parameter can be used to pass execution parameters to ADRDSSU. This parameter is intended for diagnostic purposes and can be used to gather more information when ADRDSSU has problems. The parameter values should be: ABEND=nnn [, AMSGCNT=nnnn] or SDUMP=nnn [, SMSGCNT=nnnn]. These parameters are described in the DFSMS Storage Administration Reference manual in the section "How to Control DFSMSdss through PARM Information in the EXEC Statement".

FASTREP (DSS parameter): Specifies whether fast replication is preferred (PREF, the default), required (REQ), or not used (NONE). With PREF, fast replication is used by the COPY program as long as sufficient resources are available; if not, then regular ADRDSSU COPY is used. With REQ, if sufficient resources are not available, the COPY will fail.

FCTOPPRCPPRIMARY [(PRESMIRREQ | PRESMIRPREF | PRESMIRNONE)] (DSS parameter): Indicates that a FlashCopy target volume can also be a PPRC primary volume. This applies to ESS devices only. This does not apply when FASTREP(NONE) is also specified. IBM Remote Pair FlashCopy (also known as Preserve Mirror) can be specified by including one of the optional keywords. Preserve Mirror mirrors the FlashCopy command that is issued at the local site to the remote site. This allows FlashCopy operations to occur to PPRC primary volumes without affecting the PPRC duplex state. IBM Remote Pair FlashCopy must be installed in the storage controller along with the corresponding software support in z/OS. In addition, both the source and target volumes being PPRC primary volumes and in the same storage controller and their corresponding PPRC secondary volumes being in the same storage controller. Specify one of the following use this functionality:

- **PRESMIRREQ** (short form: PMR): Require the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation will not be completed.
- **PRESMIRPREF** (short form: PMP): Prefer the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation is still performed.
- **PRESMIRNONE** (short form: PMN): Do not use Preserve Mirror.

The specified option will also be passed to ADRDSSU as part of the generated copy commands. For additional information about Preserve Mirror, refer to the documentation for your version of z/OS.

FCTOXRCPPRIMARY (DSS parameter): Indicates that a FlashCopy target volume can also be an XRC primary volume. This keyword is applicable only for DATA-MOVER PGM(ADRDSSU). This does not apply when FASTREP(NONE) is also specified.

EXCLUDE-SRCNAME-MASKS (*srcemask1, srcemask2, ... srcemaskn*)

Specifies the list of source data set name masks to exclude from the COPY-BY-DS command. Use this parameter to exclude files that would be included by RENAME-MASKS or RENAME-MASKS-DDN, but

that you do not want copied. This parameter also can be used to exclude migrated files, which cannot be copied without first being recalled.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with EXCLUDE-SRCNAME-MASKS-DDN

EXCLUDE-SRCNAME-MASKS-DDN (*ddname*)

Specifies the *ddname* of a data set containing the list of source data set name masks to exclude from the COPY-BY-DS command, as described in the EXCLUDE-SRCNAME-MASKS parameter.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with EXCLUDE-SRCNAME-MASKS

MIGRATED-DSN(ERROR | SKIP)

This option specifies how processing proceeds when a data set that matches a rename mask is not excluded by an entry in the EXCLUDE-SRCNAME-MASKS keyword, and the data set has been migrated. Migrated data sets cannot be copied by COPY-BY-DS. The MIGRATED-DSN keyword provides an alternative to the EXCLUDE-SRCNAME-MASKS keyword to exclude migrated data sets. ERROR causes the command to terminate if there are any migrated data sets that have not been excluded. SKIP causes the command to skip any migrated data sets that have not been excluded.

- Default: ERROR
- Required: No
- Restrictions: None

MGMTCLAS (*managementclass*)

Specifies the management class that replaces the source management class as input to the ACS routines.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with NULLMGMTCLAS.
- Short form: MC

NULLMGMTCLAS

Specifies that the input to the ACS routines is a null management class rather than the source data set's management class.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with MGMTCLAS.
- Short form: NMC

NULLSTORCLAS

Specifies that the input to the ACS routines is a null storage class rather than the source data set's storage class.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with STORCLAS.
- Short form: NSC

OFFLINE

This parameter specifies an OFFLINE copy, meaning that both the source and target Db2 systems must be down for the duration of the command. OFFLINE is the only supported type for COPY-BY-DS.

- Default: OFFLINE

- Required: No
- Restrictions: None

REPLACE-UNCONDITIONAL

This parameter specifies that all existing target data sets are replaced. If any target data sets exist and REPLACE-UNCONDITIONAL is not specified, COPY-BY-DS fails with return code of 8. Include this parameter with repetitive cloning scenarios, unless another method of cleaning up the target data sets is used (such as IDCAMS DELETE) before invoking COPY-BY-DS.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

For DATAMOVER(PGM(ADRDSSU)), SIMULATE specifies that COPY-BY-DS is to perform all functions except for the actual initiation of the copies and for journal inserts and updates. Specifying this parameter also generates a data set list report. The data set list report provides information about the number of commands and tasks to be generated, whether migrated data sets are found, and the names of the source and target data sets. The report is contained in messages CKZ06515I and CKZ006516I.

SIMULATE is recommended for the first time any Db2 system is cloned by using COPY-BY-DS, and whenever significant changes are made to the source Db2 system, such as the addition of new applications.

- Default: None.
- Required: No.
- Restrictions: None.

STORCLAS

Specifies the storage class that replaces the source storage class as input to the ACS routines.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with NULLSTORCLAS.
- Short form: SC

COPY-BY-DS step JCL example

This topic contains an example of COPY-BY-DS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZCPYDS.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The COPY-BY-DS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //????????? JOB , 'CKZ COPYBYS JDELETE ', CLASS=A, MSGCLASS=X
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
2 DEL HLQ?.JRNL
3 //????????? JOB , 'CKZ COPY BY DS', CLASS=A, MSGCLASS=X, NOTIFY=&SYSUID
4 //S1 EXEC PGM=CKZ00010, REGION=8M
5 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
6 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
7 //CKZPRINT DD SYSOUT=*
8 //SYSUDUMP DD SYSOUT=*
9 //JOURNAL DD DSN=HLQ?.JRNL,
// DISP=(,CATLG), UNIT=SYSALLDA,
// RECOG=KS, KEYLEN=64, KEYOFF=0,
// LRECL=600, SPACE=(CYL,(10,10))
10 //CKZIN DD *
COPY-BY-DS

```

```

7 REPLACE-UNCONDITIONAL -
8 RENAME-MASKS( -
    LB1A.BSDS0%.DATA YB1A.BSDS0%.DATA -
    LB1A.LOGCOPY%.*.DATA YB1A.LOGCOPY%.*.DATA -
    LB1A.DSNDBD.** YB1A.DSNDBD.** -
    ) -
    JOURNAL-DDN(JOURNAL) -
//*
```

1. Deletion of journal data set in anticipation of allocating new for each execution. Because this data set is used to pass information from one Db2 Cloning Tool step to another, do not delete the journal data set in any steps except the COPY-BY-DS step.
2. Execution of the Db2 Cloning Tool main program.
3. Db2 Cloning Tool SCKZLOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY-BY-DS step and used as input and output by later steps. The specified data set name must match the data set allocated in the COPY-BY-DS step. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. REPLACE-UNCONDITIONAL (the default) specifies that if existing target files are found, they are reused.
8. RENAME-MASKS supplies the source and target data set masks and defines the list of source data sets to be copied.

COPYCHECK

This command is not required. COPYCHECK is provided in case other relationships are to be initiated that require the previously initiated copies to complete, or if COPY must be rerun, and withdrawing previously established copies may save time, rather than waiting for them to finish.

COPYCHECK provides a mechanism to either 'WAIT' for copies to complete, or to 'WITHDRAW' or 'STOPSNAP' (terminate) previously established volume relationships.

- WAIT is intended for situations where other relationships need to be initiated when the Db2 Cloning Tool copies are complete.
- WITHDRAW is intended for reruns of the COPY step without waiting for copies to complete when FlashCopy is used.
- STOPSNAP is intended for reruns of the COPY step without waiting for copies to complete when EMC SNAP is used.

If the copy was established with FCNOCOPY or BACKGROUNDCOPY(NO) (no background copy), following the withdraw, some of the tracks on the volume may contain data from the source volume, while other tracks may contain residual data that was on the target volume before the copy. This situation can cause problems when trying to access the target volume if the VTOC locations of the source and target volumes were different before the copy.

COPYCHECK command syntax

```
COPYCHECK
```

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }  
WAIT( nnn | 10 [ ,RC( rr | 4 ) ] ) | WITHDRAW | STOPSNAP
```

COPYCHECK command and keyword definitions

Required keywords are described first, followed by optional keywords.

COPYCHECK

Optional command to wait for completion of COPY events, or withdraw from them.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

COPYCHECK, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs carried in the journal data set from a previously executed COPY command.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

WAIT(*nnn* | 10 [,RC(*rr* | 4)])

WAIT specifies that COPYCHECK is to check at 30-second intervals to see if copy relationships have completed for all volume copies initiated in a corresponding COPY step.

nnn specifies the maximum time in minutes that COPYCHECK should continue checking. If the specified time limit expires before all copies are complete, COPYCHECK will terminate with a return code of *rr*.

COPYCHECK will examine the volume copies that are passed via the journal data set from a corresponding COPY step.

- Default: 10,RC(4)
- Required: Yes
- Restrictions: Mutually exclusive with WITHDRAW and STOPSNAP.

WITHDRAW

WITHDRAW specifies that COPYCHECK is to withdraw all copy relationships initiated in a corresponding COPY step.

If COPY must be rerun, WITHDRAW will eliminate the otherwise required wait time for one set of volume copies to complete before copies involving the same volumes can be initiated again.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with WAIT and STOPSNAP.

STOPSNAP

STOPSNAP specifies that COPYCHECK is to stop snap all copy relationships initiated in a corresponding COPY step. If COPY must be rerun, STOPSNAP will eliminate the otherwise required wait time for one set of volume copies to complete before copies involving the same volumes can be initiated again.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with WAIT and WITHDRAW.

COPYCHECK step JCL example

This topic contains an example of COPYCHECK step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZCPYCK.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The COPYCHECK step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'CKZ COPYCHECK',CLASS=A,MSGCLASS=X
//S1 EXEC PGM=CKZ00010,REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
3 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
4 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL,DISP=OLD
//CKZIN DD *
COPYCHECK -
WAIT(20) -
JOURNAL-DDN(JOURNAL)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

DB2ALTERBSDS

This command is optional and is used to alter the contents of a target Db2 BSDS beyond what DB2UPDATE does if any of the following functionality is desired.

- Add a cold start conditional restart record in the BSDS – The keyword COLD-START will create a cold start conditional restart record in the BSDS. The RBA for the cold start will be the next highest 4k value greater than the current high written RBA.
- Delete the Db2 archive log names in the BSDS that are not on the source volumes – The keyword REMOVE-ARCHIVE-LOGS(NOTRENAMED) can remove the Db2 archive log names in the BSDS that are not on the source volumes instead of leaving them in there not renamed.

- Delete non-cloned Db2 active log names in the BSDS – The keyword REMOVE-ACTIVE-LOGS(RETAIN(n) will remove all of the active logs except n, where n is the number that is selected. The logs will be deleted in start RBA order with the log with the lowest start RBA being deleted first.
- Add a system-level backup (SLB) start conditional restart record in the BSDS – The keyword SLB-START will create an SLB start conditional restart record in the BSDS. The ENDLRSN value that is used in the conditional restart record for the SLB start comes from the System Backup record in the BSDS extracted by the prior DB2UPDATE. This option addresses situations when you are cloning from a Db2 BACKUP SYSTEM and either an active log is defined with more than one stripe or it is a data sharing group. For these cases, the active logs must be truncated at the point when the Db2 BACKUP SYSTEM FlashCopy of the database volumes completed. For data sharing, DB2ALTERBSDS SLB-START should be run for each member of the target data sharing group before the first start of the target member.

The active log will be checked to verify that the SLB LRSN exists for this Db2 subsystem. The conditional restart record will not be created if the active log does not contain the SLB LRSN. This prevents the Db2 subsystem from getting the DSNJ098E CRCR ENDLRSN OF rrr IS HIGHER THAN ANY KNOWN LRSN error message during startup when the active log does not contain the SLB LRSN. This might occur if a member of a data sharing group is infrequently used; it is possible that its active log does not contain log records that are written after the BACKUP SYSTEM data complete LRSN.

During cloning from an SLB, the DB2UPDATE command verifies that the value of the ENDLRSN of the SLB is greater or equal to the LRSN of the last checkpoint, and that the last checkpoint does not have the status "shutdown checkpoint." This information is placed in the journal file. The DB2ALTERBSDS command with the SLB-START option reads information from the journal file to verify that the SLB ENDLRSN is valid and that the backup volumes contain the Db2 SLB. If the backup volumes do not contain the Db2 SLB, then the DB2ALTERBSDS command completes with the return code that is specified in the NOT-DB2-SLB RC option. A DSNJU003 CRESTART is not issued if the NOCRCR option is specified in the NOT-DB2-SLB keyword; otherwise, DSNJU003 CRESTART is issued if the CRCR option is specified in the NOT-DB2-SLB keyword.

This command can be run only after the DB2UPDATE command has been run against the BSDS.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2ALTERBSDS command syntax

DB2ALTERBSDS

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
COLD-START
DB2-MEMBER( member-name )
DB2-NAME ( name )
NOT-DB2-SLB( RC(4 | nnnn) , CRCR | NOCRCR)
REMOVE-ACTIVE-LOGS( RETAIN( nnn ) )
REMOVE-ARCHIVE-LOGS( NOTRENAMED )
SIMULATE
SLB-START
SLB-START-NOT-CREATED( RC( @ | nnnn)
```

DB2ALTERBSDS command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2ALTERBSDS

Optional command to alter the contents of a target BSDS beyond what DB2UPDATE does.

- Required: No
- Restrictions: This command can be run only after the DB2UPDATE command has been run against the BSDS.

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

DB2ALTERBSDS, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs that are carried in the journal data set from a previously run COPY command.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

COLD-START

Specifies that a cold start conditional restart record is to be added to the BSDSs.

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command causes DB2START to automatically reply Y to the Db2 WTOR that is a result of the conditional restart record that is created by COLD-START.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with SLB-START.

DB2-MEMBER(*member-name*)

This parameter supplies the member name of the BSDS pair to be altered.

- Default: None
- Required: No
- Restrictions: This parameter is only used for Db2 data sharing.

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this Db2 subsystem or data sharing group member.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME.

Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

NOT-DB2-SLB(RC(4 | *nnnn*) , CRCR | NOCRCR)

This parameter supplies the return code and action that will be used when backup volumes do not contain the Db2 system-level backup (SLB). Db2 Cloning Tool checks the value of the ENDLRSN of the SLB. If it is less than the LRSN of the last checkpoint, or if the last checkpoint has the status "shutdown checkpoint," the backup volumes do not contain the Db2 SLB. If NOCRCR is specified, a conditional restart record is not created for the Db2 subsystem when the backup volumes do not contain the Db2 SLB. If CRCR is specified, a conditional restart record is created for the Db2 subsystem. Valid values for *nnnn* are 0 - 4095.

- Default: RC(4), NOCRCR
- Required: No
- Restrictions: None

REMOVE-ACTIVE-LOGS(RETAIN(*nnn*))

Specifies that active logs are to be removed from the BSDSs.

RETAIN(*nnn*) specifies that all but *nnn* active logs will be removed from the BSDS.

The active logs are removed in start RBA sequence where the active log with the lowest RBA will be deleted first. Care should be taken so there are enough active logs in the BSDS so that Db2 starts up.

- Default: None
- Required: No
- Restrictions: None

REMOVE-ARCHIVE-LOGS(NOTRENAMED)

Specifies that archive logs are to be removed from the BSDSs.

NOTRENAMED specifies that archive logs that were not renamed will be removed.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that the actions alter the BSDS are printed as control cards to the Db2 DSNJU003 utility, but no modifications are made to the target BSDSs.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

SLB-START

Specifies that a system-level backup (SLB) start conditional restart record is to be added to the BSDS. The SLB start conditional restart record that is created will have an ENDLRSN value that comes from the system backup record in the BSDS that was extracted by the prior DB2UPDATE.

SLB-START is needed when you are cloning from a Db2 BACKUP SYSTEM and either an active log is defined with more than one stripe or it is a data sharing group. For these cases, the active logs must be truncated at the point when the Db2 BACKUP SYSTEM FlashCopy of the database volumes completed. For data sharing, DB2ALTERBSDS SLB-START should be run for each member of the target data sharing group before the first start of the target member.

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command causes DB2START to automatically reply Y to the Db2 WTOR that is a result of the conditional restart record that is created by SLB-START.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with COLD-START.

SLB-START-NOT-CREATED(RC(0 | nnnn))

This parameter supplies the return code that will be used if the active log of this Db2 subsystem does not contain the LRSN of the SLB that is being used for a conditional restart, and a conditional restart record will not be created for this Db2 subsystem. Valid values for *nnnn* are 0 - 4095.

- Default: RC(0)
- Required: No
- Restrictions: None

DB2ALTERBSDS step JCL example

Three DB2ALTERBSDS step JCL examples are included: an example to create cold start conditional restart record, an example to remove not renamed archive logs, and an example to remove some active logs. Sample JCL can be found in the installation library SCKZJCL in member CKZDALBS.

DB2ALTERBSDS Step JCL – example to create cold start conditional restart record

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2ALTERBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'DB2ALTERBSDS' , CLASS=A ,MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010 ,REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD ,DISP=SHR
4 // DD DSN=DSNxxx.SDSNLOAD ,DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) ,DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SORTMSG DD SYSOUT=*
8 //SYSUDUMP DD SYSOUT=*
9 //JOURNAL DD DSN=CKZ.JRNL ,DISP=OLD
10 //CKZIN DD *
11 DB2ALTERBSDS -
12 COLD-START -
13 DB2-MEMBER(DB1T) -
14 JOURNAL-DDN(JOURNAL)
15 //*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. COLD-START parameter that specifies a cold start conditional restart records is to be created.

8. DB2-MEMBER parameter specifying Db2 data sharing group member name, DB1T, is to have its BSDSs updated.

DB2ALTERBSDS Step JCL – example to remove not renamed archive logs

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2ALTERBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

  //??????? JOB , 'DB2ALTERBSDS' , CLASS=A, MSGCLASS=X
1 //S1 EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 // DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SORTMSG DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN DD *
  DB2ALTERBSDS -
7 REMOVE-ARCHIVE-LOGS(NOTRENAMED) -
8 DB2-MEMBER(DB1T) -
6 JOURNAL-DDN(JOURNAL)
  //*

```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. REMOVE-ARCHIVE-LOGS parameter that specifies all not renamed archive log records be removed.
8. DB2-MEMBER parameter specifying Db2 data sharing group member name, DB1T, is to have its BSDSs updated.

DB2ALTERBSDS Step JCL – Example to Remove Some Active Logs

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2ALTERBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

  //??????? JOB , 'DB2ALTERBSDS' , CLASS=A, MSGCLASS=X
1 //S1 EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 // DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SORTMSG DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN DD *
  DB2ALTERBSDS -
7 REMOVE-ACTIVE-LOGS(RETAIN(3)) -

```

```
8      DB2-MEMBER(DB1T)
6      JOURNAL-DDN(JOURNAL)
      //*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for Db2 Cloning Tool.SCKZPARM, CKZINI member. The CKZINI member is used to provide variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. REMOVE-ACTIVE-LOGS parameter that specifies to remove all but the most recent 3 active logs.
8. DB2-MEMBER parameter specifying Db2 data sharing group member name, DB1T, is to have its BSDSs updated.

DB2FIX

This command is not required. This command is only used for 'online' cloning of a Db2 subsystem. An online cloning is when the source Db2 subsystem is active (online) at the time the source volumes were copied. DB2FIX will fix target Db2 page spaces that have LPL or GRECP status by issuing a Db2 START DATABASE command against them. If the Db2 system is data sharing, only one Db2 member should be running when DB2FIX is run.

The DB2FIX command should be run twice. The first run will fix any Db2 CATALOG (DSNDB06) or Db2 Directory (DSNDB01) page spaces by using DATABASES(Db2) and the second run will fix all other page spaces by using DATABASES(APPLICATION). The second run with DATABASES(APPLICATION) must only happen after the Db2 catalog has been updated with the DB2SQL command.

When using the keyword DATABASES(Db2), if page space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was started due to having LPL or GRECP status, DB2FIX will: 1) complete with the return code specified in the DSNDB01-DBD01-STARTED, MEMBERS-AND-DBD01, or WAIT-AND-DBD01 keywords, and 2) display the following WTO message:

```
CKZ23526E DSNDB01.SYSDBDXA IS IN RESTRICTED STATUS; DB2UPDATE NEEDS TO BE RUN AGAIN
```

For Db2 10 CM the message is:

```
CKZ23526E DSNDB01.DBD01 IS IN RESTRICTED STATUS; DB2UPDATE NEEDS TO BE RUN AGAIN
```

If this error happens, the changes made to SYSDBDXA (DBD01 for Db2 10 CM) by DB2UPDATE may have been regressed and need to be redone. The target Db2 subsystem must be stopped, DB2UPDATE run again using the DBD01ONLY keyword, and the Db2 subsystem started again in maintenance mode using the special zparms, prior to running DB2SQL and DB2FIX using DATABASES(APPLICATION).

These steps can be automated by using either the DB2FIX return code setting, or have an automated operations package monitor the DB2FIX WTO message produced when DB2FIX starts restricted page spaces.

If not all data is being cloned with the Db2 subsystem, it is possible that some of the table and index spaces that are not cloned will have LPL or GRECP status. DB2FIX will receive a timeout on the start commands issued for these page spaces. The EXCLUDE-MASKS keyword can be used to cause DB2FIX to not issue start commands for the page spaces that are not cloned.

For data sharing, there is a possibility that other members will hold locks on table and index spaces that have LPL or GRECP status. Normally the locks are released when the other members are initially started in special and light mode. If some locks are not released by a member after the initial start, it will be necessary to start that member in special and not light mode so it remains running. Then running DB2FIX with MEMBERS-NEED-STARTING(ACTION(CONTINUE)) will usually resolve the LPL or GRECP status. If DB2FIX is still unable to resolve the LPL or GRECP status, then running DB2FIX with MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on each active member will usually resolve the LPL or GRECP status.

The user ID that runs the DB2FIX command must be defined in the special zparms as either SYSADM or SYSADM2.

For more information on this command, see the following topics:

- [“Status of transactions in flight” on page 135](#)
- [“Db2 online cloning with removal of data sharing members procedure” on page 140](#)
- [“Db2 online cloning with target becoming non-data sharing procedure” on page 146](#)

DB2FIX command syntax

DB2FIX

Required keywords:

```
DATABASES( DB2 | APPLICATION )
DB2-SSID( db2 ssid )
```

Optional keywords:

```
DSNDB01-DBD01-STARTED( RC( nn | 16 ) )
EXCLUDE-MASKS( masks )
MAX-CONCURRENT-CMDS( nn | 1 )
MEMBERS-AND-DBD01( RC( nn | 16 ) )
MEMBERS-NEED-STARTING( [ RC( nn | 8 ) ] [ACTION( QUIT | CONTINUE ) ] )
SIMULATE
START-SCOPE( DATABASE | PAGESPACE )
WAIT( nnn | 5 [,RC( rr | 8 ) ] [ACTION( QUIT | CONTINUE ) ] )
WAIT-AND-DBD01( RC( nn | 16 ) )
```

EXCLUDE-MASKS Considerations

The exclude mask is used to select the table and index spaces that will not be started by DB2FIX. The format of an exclude mask entry is dbname.spname, where dbname is the database name and spname is the space name. The allowable filter characters are shown in the following table:

| Character | Description |
|-----------|---|
| * | A single asterisk represents 0 to nn characters of any value. |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| < | A lesser-than sign represents one non-numeric character, national symbols included. |
| > | A greater-than sign represents one numeric character. |

For example, exclude mask = APPL1DB.* would match all spaces in database APPL1DB.

For information about filters and ACS masks, refer to the topic [“Filtering pattern masks” on page 26](#).

DB2FIX command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2FIX

Optional command to correct Db2 page spaces that are in restricted status.

- Required: No
- Restrictions: This command is only used for 'online' cloning of a Db2 subsystem with Db2 SET LOG SUSPEND.

DATABASES(Db2 | APPLICATION)

This parameter specifies which database page spaces will be fixed.

Db2 specifies that page spaces in the Db2 catalog (DSNDB06) and Db2 Directory (DSNDB01) are to be fixed.

APPLICATION specifies that page spaces in all other databases (not DSNDB01 and DSNDB06) are to be fixed. The running of DB2FIX with DATABASES(APPLICATION) must only be done after the running of the DB2SQL command.

- Default: None
- Required: Yes
- Restrictions: None

DB2-SSID(db2 ssid)

This parameter supplies the Db2 SSID of the Db2 subsystem to connect to. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

DSNDB01-DBD01-STARTED(RC(nn | 16))

This parameter supplies the return code that will be used if table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was started due to the table space having LPL or GRECP status.

- Default: RC(16)
- Required: No
- Restrictions: None.

EXCLUDE-MASKS(masks)

This parameter supplies a list of masks of the form dbname.spname, where dbname is the database name and spname is the space name. If a database and page space name matches an entry in the list, no start command will be issued for it.

If START-SCOPE(DATABASE) is also specified, page space names that match an entry in the EXCLUDE-MASKS list will still be started if there are page spaces in the same database that are not in the list.

- Default: None
- Required: No
- Restrictions: Applies only when DATABASES(APPLICATION) is specified.

MAX-CONCURRENT-CMDS(nn | 1)

Specifies the maximum number of start commands that will be processed concurrently. Increasing this number can decrease the elapsed time of DB2FIX and will increase the resources being used by Db2. Experiment with this parameter to determine the optimum setting.

- Default: 1
- Required: No
- Restrictions: Applies only when DATABASES(APPLICATION) is specified.

MEMBERS-AND-DBD01(RC(nn | 16))

This parameter supplies the return code that will be used if table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSNDB01 was started (due to the table space having LPL or GRECP status), and if other members in this data sharing group need to be started because they hold locks.

- Default: RC(16)
- Required: No
- Restrictions: None

MEMBERS-NEED-STARTING([RC(nn | 8)] [, ACTION(QUIT | CONTINUE)])

This parameter supplies the return code that will be used and the action to be taken if other members in this data sharing group need to be started because they hold locks. The ACTION parameter specifies whether to QUIT processing or to CONTINUE processing by attempting to start restricted objects when locks are held by other members.

ACTION(CONTINUE) is used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using ACTION(CONTINUE), it may be necessary to run DB2FIX using ACTION(CONTINUE) on the other members as well. This option addresses the situation for data sharing where other members are reported to hold locks even after they have been started in light mode.

- Default: RC(8), ACTION(QUIT)
- Required: No
- Restrictions: None

SIMULATE

Specifies that DB2FIX will show what it will do, but will not issue any Db2 START DB commands.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

START-SCOPE(DATABASE | PAGESPACE)

Specifies the scope of the Db2 start database commands that will be used.

DATABASE specifies that a single Db2 start database command will be done for all page spaces in a database that have LPL or GRECP status (e.g. STA DB(dbname) SPACENAM(*)).

PAGESPACE specifies that a single Db2 start database command will be done for each page space in the database that has LPL or GRECP status (e.g. STA DB(dbname) SPACENAM(spacename)).

- Default: PAGESPACE
- Required: No
- Restrictions: Applies only when DATABASES(APPLICATION) is specified.

WAIT(nnn | 5 [, RC(rr | 8)] [, ACTION(QUIT | CONTINUE)])

nnn specifies the maximum time in minutes that DB2FIX will wait for a single Db2 start database command to fix the page spaces that were in LPL or GRECP status. If the specified time limit expires before the Db2 start database command has fixed the page spaces in LPL or GRECP status, DB2FIX will terminate with return code rr. When a timeout occurs, the ACTION parameter specifies whether to QUIT processing or CONTINUE processing.

- Default: 5,RC(8), ACTION(QUIT)
- Required: No
- Restrictions: ACTION(CONTINUE) applies only when DATABASES(APPLICATION) is specified.

WAIT-AND-DBD01(RC(nn | 16))

This parameter supplies the return code that will be used if table space SYSDBDXA (DBD01 for Db2 10 CM) in database DSND01 was started due to its being found to have LPL or GRECP status and if the WAIT time limit was exceeded for a Db2 start database command to fix page spaces in LPL or GRECP status.

- Default: RC(16)
- Required: No
- Restrictions: None.

DB2FIX step JCL example

This topic contains an example of DB2FIX step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDFIX.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2FIX step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2FIX' , CLASS=A , MSGCLASS=X
//S1 EXEC PGM=CKZ00010 , REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD , DISP=SHR
3 // DD DSN=DSNxxx . SDSNLOAD , DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) , DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CKZIN DD *
DB2FIX -
6 DB2-SSID(DB1T) -
7 DATABASES(DB2)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying Db2 ssid, DB1T, that will be connected to.
7. DATABASES parameter specifying the databases, Db2, that will be checked for LPL and GRECP status and started as necessary.

DB2GETBACKINFO

This command is not required. DB2GETBACKINFO will issue HSM LIST COPYPOOL commands to determine the available backups from Db2 BACKUP SYSTEM and read the results to determine the source and backup volume pairs to be used. The source volume to backup volume pairs and optionally source ICF catalog information will be written to the backinfo data set.

This command is used as part of the process to clone from a Db2 BACKUP SYSTEM backup or backup dump tapes. Detailed descriptions of these cloning processes can be found in [“Cloning scenarios” on page 1091](#).

Using the LIST option will cause information about all backups or a specific backup to be displayed for a specific Db2 system.

For a backup that was created using Db2 BACKUP SYSTEM, the LOCATION or BSDS-DDN keywords are used to specify the location to be used in the copypool names. For a backup that was created by directly

using HSM FRBACKUP, the DB-COPYPOOL-NAME and LG-COPYPOOL-NAME keywords are used to specify the full copypool names to be used.

Information about backups is obtained from HSM by issuing commands in one of two ways:

- When LOCATION or BSDS-DDN are used:

```
LIST COPYPOOL(DSN$location$type) ODS(workdsn)
```

Where:

- *type* is DB for the database copy pool or LG for the log copy pool.
 - The *location* used comes from the LOCATION keyword, or is read from a BSDS if the BSDS-DDN keyword is used.
 - The *workdsn* data set name comes from the WORK-DDN or WORK-DSN keyword. This data set cannot be a temporary data set.
- When DB-COPYPOOL-NAME and LG-COPYPOOL-NAME are used:

```
LIST COPYPOOL(copypoolname) ODS(workdsn)
```

Where:

- *copypoolname* is the value specified for DB-COPYPOOL-NAME or LG-COPYPOOL-NAME.
- The *workdsn* data set name comes from the WORK-DDN or WORK-DSN keyword. This data set cannot be a temporary data set.

HSM writes the output of the list command to the workdsn data set. DB2GETBACKINFO then reads and parses the data set to identify the backups and their information.

The use of the HSM LIST COPYPOOL command can be protected by RACF with a profile in the FACILITY class of the form STGADMIN.ARC.LC.copypoolname. The userid that runs the DB2GETBACKINFO command must have at least READ access to such a profile if it exists.

The work data set identified by the WORK-DDN or WORK-DSN keyword is updated by both HSM and the DB2BACKINFO command. HSM and the user ID that runs the DB2GETBACKINFO command must have at least UPDATE authority to this data set.

DB2GETBACKINFO command syntax

DB2GETBACKINFO

Required keywords:

```
{ BSDS-DDN( ddname ) | LOCATION( location ) |  
  DB-COPYPOOL-NAME( dbcpname )  
  LG-COPYPOOL-NAME( lgcpname ) }  
{ WORK-DDN( ddname ) | WORK-DSN( dsname ) }
```

Optional keywords:

```
BACKINFO-DDN( ddname )  
{ TOKEN( X'token' ) | LAST }  
LIST  
{ LOGONLY | DATABASESONLY }  
USE-DUMPTAPES  
DUMP-CLASS( classname )  
USERCATALOGS( sourcecat1, ..., sourcecatn )
```

DB2GETBACKINFO command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2GETBACKINFO

Optional command to allow for easier cloning from a Db2 BACKUP SYSTEM FULL backup.

- Required: No.
- Restrictions: None.

BSDS-DDN(*ddname*)

This parameter specifies the DD name that points to a BSDS from which the location will be obtained. The location will be used in the HSM copy pool names.

- Default: None.
- Required: Yes.
- Restrictions: Mutually exclusive with LOCATION, DB-COPYPOOL-NAME, and LG-COPYPOOL-NAME.

LOCATION(*location*)

This parameter specifies the location that will be used in the HSM copy pool names.

- Default: None.
- Required: Yes.
- Restrictions: Mutually exclusive with BSDS-DDN, DB-COPYPOOL-NAME, and LG-COPYPOOL-NAME.

DB-COPYPOOL-NAME(*dbcpname*)

This parameter specifies the database (DB) volumes backup copypool name that will be used in the HSM LIST COPYPOOL command.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with BSDS-DDN and LOCATION.

LG-COPYPOOL-NAME(*lgcpname*)

This parameter specifies the log (LG) volumes backup copypool name that will be used in the HSM LIST COPYPOOL command.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with BSDS-DDN and LOCATION.

WORK-DDN(*ddname*)

This parameter specifies the DD name that points to the HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command. The data set allocated to this DD cannot be a temporary data set.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with WORK-DSN.

WORK-DSN(*dsname*)

This parameter specifies the data set name of the HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command.

- Default: None.
- Required: Yes.
- Restrictions: Mutually exclusive with WORK-DDN.

BACKINFO-DDN(*ddname*)

This parameter specifies the DD name which points to a file where the backinfo information will be written.

- Default: None.
- Required: Required if LIST is not specified.
- Restrictions: Not used if LIST is specified.

DATABASESONLY

This parameter specifies that only the database (DB) volumes of the selected backup should be used. DATABASESONLY is intended to be used when you want to clone only the database volumes of the selected backup.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with LOGSONLY.
- Short form: DBONLY

DUMP-CLASS(*classname*)

This parameter specifies the tape dump class to identify the correct tape volumes. This option addresses the situation when there are multiple dump classes for a Db2 BACKUP SYSTEM.

- Default: None.
- Required: Only required if USE-DUMPTAPES is specified and there is more than one tape dump class used for the backup.
- Restrictions: Only valid if USE-DUMPTAPES is specified.

LAST

This parameter specifies that the last (most recent) backup is to be used.

- Default: None.
- Required: Required if LIST is not specified.
- Restrictions: Mutually exclusive with TOKEN.

LIST

This parameter specifies that a list be generated. If LAST or TOKEN are also specified only one backup, corresponding to the last or the token value, will be listed.

- Default: None.
- Required: No.
- Restrictions: None.

LOGSONLY

This parameter specifies that only the log (LG) volumes of the selected backup should be used. LOGSONLY is intended to be used when you want to clone only the log volumes of the selected backup.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with DATABASESONLY.
- Short form: LGONLY

TOKEN(*X'token'*)

This parameter specifies that a specific backup is to be used. The value of token to use can be obtained by using the LIST option of this command, using a HSM LIST COPYPOOL command, or running the DSNJU004 utility.

- Default: None.
- Required: Required if LIST is not specified.
- Restrictions: Mutually exclusive with LAST.

USE-DUMPTAPES

Specify this parameter to retrieve information about system-level backup dump tape volumes instead of the backup volumes. The information from HSM is used to populate the backinfo data set.

When you specify USE-DUMPTAPES, DB2GETBACKINFO excludes the following from consideration:

- Backups with an HSM DUMPSTATE that is not ALLCOMPLETE
- Backups with at least one dump tapes that have an expiration date prior to the current date
- Default: None.
- Required: No.
- Restrictions: None.

USERCATALOGS(*srcat1*, ..., *srcatn*)

This parameter specifies the source ICF catalogs that are being used by the source Db2 system. It is required that all the specified ICF catalogs reside on source volumes.

- Default: None.
- Required: No.
- Restrictions: None.

DB2GETBACKINFO step JCL examples

This topic contains examples of DB2GETBACKINFO step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2GETBACKINFO step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'CKZ DB2GETBACKINFO' , CLASS=A, MSGCLASS=X
2 //S0 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 1 DEL CKZ.WRK.BACKINFO
6 2 DEL CKZ.WRK.HSMLIST
7 SET MAXCC=0
8 3 //S1 EXEC PGM=CKZ00010, REGION=8M
9 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
10 //CKZINI DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
11 //CKZPRINT DD SYSOUT=*
12 //SYSUDUMP DD SYSOUT=*
13 7 //BACKINFO DD DSN=CKZ.WRK.BACKINFO,
14 // DISP=(,CATLG), UNIT=SYSALLDA,
15 // SPACE=(CYL,(1,1))
16 8 //HSMLIST DD DSN=CKZ.WRK.HSMLIST,
17 // DISP=(,CATLG), UNIT=SYSALLDA,
18 // SPACE=(CYL,(1,1))
19 //CKZIN DD *
20 DB2GETBACKINFO -
21 7 BACKINFO - DDN(BACKINFO) -
22 8 WORK - DDN(HSMLIST) -
23 9 LAST -
24 10 LOCATION(DB2PLOC) -
25 11 USERCATALOGS( -
26 USERCAT.SRC01 -
27 USERCAT.SRC02 -
28 )
29 /**
```

1. Deletion of backinfo data set in anticipation of allocating new for each execution.
2. Deletion of HSM list data set in anticipation of allocating new for each execution.
3. Execution of Db2 Cloning Tool main program.
4. Db2 Cloning Tool LOAD library must be authorized.

5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
6. DD for CKZPRINT output.
7. The backinfo data set that is created by DB2GETBACKINFO and will be used by a subsequent Db2 Cloning Tool BACKINFO-REFORMAT command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used.
8. The HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command. In the sample JCL, the control statement WORK-DDN(HSMLIST) specifies that a DD statement with the name HSMLIST is used to identify the work data set. This data set cannot be a temporary data set.
9. Specifies that the last Db2 BACKUP SYSTEM backup be used.
10. The location that identifies the Db2 whose BACKUP SYSTEM backup is to be used.
11. The source ICF catalogs that are being used by the source Db2 system. These ICF catalogs must reside on source volumes.

Example of using DB2GETBACKINFO to get BACKUP SYSTEM dump tape information

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

The DB2GETBACKINFO step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'CKZ DB2GETBACKINFO', CLASS=A, MSGCLASS=X
2 //S0 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 1 DEL CKZ.WRK.BACKINFO
6 2 DEL CKZ.WRK.HSMLIST
7 SET MAXCC=0
8 //S2 EXEC PGM=CKZ00010, REGION=8M, PARM= ' '
9 //*
10 //STEPLIB DD DISP=SHR, DSN=HLQ?.LOADLIB
11 //CKZINI DD DISP=SHR, DSN= HLQ?.SCKZPARM(CKZINI)
12 //CKZPRINT DD SYSOUT=*
13 //BACKINFO DD DISP=(,CATLG), DSN=HLQ?.BACKINFO,
14 // UNIT=SYSALLDA, SPACE=(TRK,(3,3))
15 8 //HSMLIST DD DISP=(,CATLG), DSN=HLQ?.HSMLIST,
16 // UNIT=SYSALLDA, SPACE=(TRK,(3,3))
17 //CKZIN DD *
18 DB2GETBACKINFO -
19 USE-DUMPTAPES -
20 DUMP-CLASS(DUMPTP) -
21 7 BACKINFO-DDN(BACKINFO) -
22 8 WORK-DDN(HSMLIST) -
23 11 LAST -
24 12 LOCATION(DB2-LOCATION) -
25 13 USERCATALOGS(
26 USERCAT.SRC1 -
27 USERCAT.SRC2 -
28 )
29 //*

```

1. Deletion of backinfo data set in anticipation of allocating new for each execution.
2. Deletion of HSM list data set in anticipation of allocating new for each execution.
3. Execution of Db2 Cloning Tool main program.
4. Db2 Cloning Tool LOAD library must be authorized.
5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
6. DD for CKZPRINT output.
7. The backinfo data set that is created by DB2GETBACKINFO and will be used by a subsequent Db2 Cloning Tool RESTORE-FROM-DUMPTAPES command. In the sample JCL, the control statement

BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used. The backinfo data set is input to the command and must be generated by a DB2GETBACKINFO command from the source Db2 system or LPAR. If not on shared DASD with the target system, the backinfo data set must be transferred via FTP or other method to the target system.

8. The HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command. In the sample JCL, the control statement WORK-DDN(HSMLIST) specifies that a DD statement with the name HSMLIST is used to identify the work data set. This data set cannot be a temporary data set.
9. Directs DB2GETBACKINFO to look only for dump tapes in the HSM LIST COPYPOOL output. This keyword is required when generating a backinfo data set for a RESTORE-FROM-DUMPTAPES command.
10. Directs DB2GETBACKINFO to search only for the specified DUMP-CLASS in the HSM LIST COPYPOOL output. Replace *DUMPTP* with the actual DUMP-CLASS. This keyword is optional; if not specified and multiple tape dump classes are found, the first one found in the HSM LIST COPYPOOL output is used.
11. Specifies that the last Db2 BACKUP SYSTEM backup be used.
12. The location that identifies the Db2 whose BACKUP SYSTEM backup is to be used.
13. The source ICF catalogs that are being used by the source Db2 system. These ICF catalogs must reside on source volumes.

DB2LGRNXCLEAN

This command is optional. It is used only for the cloning of a Db2 data sharing group when you want to physically remove members from the target Db2 subsystem.

DB2LGRNXCLEAN removes all entries from SYSLGRNX. For more information about this command, see the following topics:

- [“Db2 offline cloning with removal of data sharing members procedure” on page 128](#)
- [“Db2 online cloning with removal of data sharing members procedure” on page 140](#)

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2LGRNXCLEAN command syntax

```
DB2LGRNXCLEAN
```

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
DB2-NAME ( name )  
SIMULATE
```

DB2LGRNXCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2LGRNXCLEAN

Optional command to clean out Db2 SYSLGRNX as part of removing other members.

- Required: No

- Restrictions: Used only for the cloning of a Db2 data sharing group when you want to physically remove members from the target Db2 subsystem.

**JOURNAL-DSN (*data set name*)
or JOURNAL-DDN (*ddname*)**

This parameter supplies either the data set name of the Db2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

DB2ALTERBSDS, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs that are carried in the journal data set from a previously run COPY command.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this Db2 subsystem or data sharing group.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME.

Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that what will be done to clean out Db2 SYSLGRNX is shown, but no modifications are made to SYSLGRNX.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2LGRNXCLEAN step JCL example

This topic contains an example of DB2LGRNXCLEAN step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDLGCL.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2LGRNXCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'DB2LGRNXCLEAN', CLASS=A, MSGCLASS=X
//S1 EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
// DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
3 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
//CKZIN DD *
DB2LGRNXCLEAN
JOURNAL-DDN(JOURNAL)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

DB2RBLDBSDS

This command is optional. This command is used only for the cloning of a Db2 data sharing group when you want to physically remove members from the target BSDS or to make the target Db2 non-data sharing.

This command must be run for every target member that is to be used. For more information about this command, see the following topics:

- [“Db2 offline cloning with removal of data sharing members procedure” on page 128](#)
- [“Db2 offline cloning with target becoming non-data sharing procedure” on page 131](#)
- [“Db2 online cloning with removal of data sharing members procedure” on page 140](#)
- [“Db2 online cloning with target becoming non-data sharing procedure” on page 146](#)

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2RBLDBSDS command syntax

DB2RBLDBSDS

Required keywords:

```

DB2-MEMBER( member name )
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
ACTIVELOG-LIMIT( ALL | nnnn )  
DATA-SHARING( SOURCE | NO )  
DB2-NAME ( name )  
DDF( SOURCE | NO )  
PRIMARY( SOURCE | YES | NO )  
SIMULATE
```

DB2RBLDBSDS command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2RBLDBSDS

Optional command to rebuild a Db2 BSDS to remove other members or to make it non-data sharing.

- Required: No
- Restrictions: Used only for cloning a Db2 data sharing group when you want to physically remove members from the target BSDS or to make the target Db2 non-data sharing.

DB2-MEMBER (*member name*)

This parameter supplies the member name of the BSDS pair to be rebuilt.

- Default: None
- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

DB2ALTERBSDS, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs that are carried in the journal data set from a previously run COPY command.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

ACTIVELOG-LIMIT (ALL | *nnnn*)

Specifies how many active logs will be retained in the rebuilt target BSDS for this member. ALL specifies that all active logs be retained in the BSDS. *nnnn* specifies the number of active logs to retain in the BSDS. Care should be taken so there are enough active logs in the BSDS so that Db2 starts up.

- Default: ALL
- Required: No
- Restrictions: This parameter is not used if DATA-SHARING(NO) is specified.

DATA-SHARING (SOURCE | NO)

SOURCE means that the actions to alter the BSDS are printed as control cards to the Db2 DSNJU003 utility, but no modifications are made to the target BSDSs. The DSNJU003 utility with DATASHR ENABLE, NEWCAT, and NEWLOG (if needed) is invoked. The DSNJCNVB utility is invoked if the subsystem must be converted to use extended RBA or needs to use more log data sets. NO means

that the specified target subsystem becomes non-data sharing. The DSNJU003 utility with DATASHR DISABLE and CRESTART CREATE is invoked.

- Default: Source
- Required: No
- Restrictions: If DATA-SHARING(NO) is specified, keywords ACTIVELOG-LIMIT, DDF, and PRIMARY are not used.

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this Db2 subsystem or data sharing group member.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME.

Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

DDF (SOURCE | NO)

Specifies whether the DDF information in the rebuilt BSDS will be retained for this member. SOURCE specifies that the DDF information is retained. NO specifies that the DDF information is not retained.

- Default: Source
- Required: No
- Restrictions: This parameter is not used if DATA-SHARING(NO) is specified.

PRIMARY (SOURCE | YES | NO)

Specifies whether the rebuilt target BSDS for this member will be for the primary of the Db2 data sharing group. SOURCE specifies that the primary BSDS will remain as the primary. YES specifies that this will be the primary member. NO specifies that this will not be the primary member.

- Default: Source
- Required: No
- Restrictions: This parameter is not used if DATA-SHARING(NO) is specified.

SIMULATE

Specifies that the actions to alter the BSDS are printed as control cards to the Db2 DSNJU003 utility but no modifications are made to the target BSDSs.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2RBLDBSDS step JCL example

This topic contains an example of DB2RBLDBSDS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDRBBS.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2RBLDBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1  //??????? JOB , 'DB2RBLDBSDS' ,CLASS=A,MSGCLASS=X
   //S1 EXEC PGM=CKZ00010,REGION=8M
2  //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
   // DD DSN=DSNxxx.SDSNLOAD,DISP=SHR
3  //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
4  //CKZPRINT DD SYSOUT=*
   //SYSUDUMP DD SYSOUT=*
5  //JOURNAL DD DSN=CKZ.JRNL,DISP=OLD
   //CKZIN DD *
   DB2RBLDBSDS -
6  DB2-MEMBER(DB1T) -
   JOURNAL-DDN(JOURNAL)
   //*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
6. DB2-MEMBER parameter specifying the member, DB1T, whose BSDS data sets will be rebuilt.

DB2SCHEMA-UPDATE

This command is not required. The DB2SCHEMA-UPDATE command can be used to change schema values in the catalog of the cloned subsystem(s).

After DB2SCHEMA-UPDATE checks input schema pairs for restrictions (see below), DB2SCHEMA-UPDATE generates DROP and CREATE statements for objects that cannot be processed by the Db2 CATMAINT utility. These objects are views, materialized query tables, SQL scalar functions, triggers, and native SQL procedures.

DROP statements are executed immediately after generation. For the rest of the objects, Db2 Cloning Tool invokes the Db2 CATMAINT utility to change schema values. After that, the generated CREATE statements are run to create views, materialized query tables, SQL scalar functions, triggers, and native SQL procedures with translated schema values. All objects use the new schema values that you specify with the DB2SCHEMA-UPDATE command.

The following types of objects cannot be processed by DB2SCHEMA-UPDATE or the CATMAINT utility. If these objects are found during the DB2SCHEMA-UPDATE run, the job finishes with an error and the job output lists the problematic objects:

- Expression-based indexes
- Column masks
- Row permissions

To avoid this error, search for these types of objects before running the DB2SCHEMA-UPDATE step. If they are found, delete the objects before running the DB2SCHEMA-UPDATE step, and then re-create the objects after the DB2SCHEMA-UPDATE step runs.

To re-create objects on the target Db2 subsystem, the Db2 Cloning Tool DB2SCHEMA-UPDATE command uses SQL and has a plan and package that must be installed. For more information about installing the plan and package, see the topic [“Cloning a Db2 subsystem” on page 112](#).

The new mask values for schema pairs should specify schema values that do not exist on the target subsystem. In addition, applying the new mask values should not result in more than one source schema value being translated to one target schema value. If this occurs, the Db2 CATMAINT utility run finishes with an error.

It is recommended to first run the DB2SCHEMA-UPDATE command with the SIMULATE keyword to check for objects with restrictions, for schema pairs with duplicate new schema values, and for new schema values that already exist on the target subsystem. If the simulation successfully completes, the command can be run without simulation.

The user ID that runs the DB2SCHEMA-UPDATE command must be defined in the special zparms as either SYSADM or SYSADM2.

The DB2SCHEMA-UPDATE command can be run only after the DB2SQL command has been run for offline cloning, or after the DB2FIX DATABASES(APPLICATION) command has been run for online cloning.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2SCHEMA-UPDATE command syntax

DB2SCHEMA-UPDATE

Required Keywords:

```
CREATE-DDL-DDN ( ddname )
DB2-SSID( db2 ssid )
DROP-DDL-DDN ( ddname )
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
SCHEMA-MASKS( mask pairs )
```

Optional Keywords:

```
DB2-NAME ( name )
EXECUTE-CREATE-DDL ( YES | NO )
RESUME
SIMULATE
USE-DDL-SQLID ( sqlid )
```

SCHEMA-MASKS Considerations

Oldvalue Syntax

The old value filter mask is used to select the schema values against which to apply the new value mask. The allowable filter characters are shown in the following table:

| Character | Description |
|-----------|---|
| * | A single asterisk represents 0 to <i>nn</i> characters of any value. |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character: @ # \$. |
| < | A lesser-than sign represents one non-numeric character, national symbols included. |
| > | A greater-than sign represents one numeric character. |

For example, old value filter mask = D%2S* would match schema value = DB2SCHM1.

For information about filters and ACS masks, refer to the topic [“Use of the U.S.A. EBCDIC code set”](#) on page 25.

Newvalue Syntax

The new value filter mask is used to rename the schema values that are selected by the old value filter mask. The allowable filter characters are shown in the following table:

| Character | Description |
|------------------|---|
| * | A single asterisk represents 0 to nn characters of any value. The single asterisk may only be used as the last character of the mask. |
| % | A percent sign represents one non-blank character. |

For example: Original schema value: DB2OSCH1; new value filter mask = TS%N*; new schema value: TS2NSCH1.

For information about filters and ACS masks, refer to the topic [“Use of the U.S.A. EBCDIC code set”](#) on page 25.

DB2SCHEMA-UPDATE command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2SCHEMA-UPDATE

Optional command to generate and run the SQL statements that are necessary to re-create some types of objects that cannot be processed by the Db2 CATMAINT utility with new schema values, and to invoke the Db2 CATMAINT utility to change schema values for the rest of the objects.

- Required: No
- Restrictions: None

CREATE-DDL-DDN (ddname)

This parameter supplies the ddname assumed via JCL to point at the data set that will be used to store the DDL generated by Db2 Cloning Tool Subsystem Cloning for creating objects that cannot be processed by the Db2 CATMAINT utility with new schemas. These objects include views, materialized query tables, SQL scalar functions, triggers, and native SQL procedures. The data set pointed to by CREATE-DDL-DDN must have an LRECL of 80 and RECFM of FB.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: CRDDN

DB2-SSID(db2-ssid)

This parameter supplies the Db2 SSID of the Db2 subsystem to connect to. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

DROP-DDL-DDN (ddname)

This parameter supplies the ddname assumed via JCL to point at the data set that will be used to store the DDL generated by Db2 Cloning Tool Subsystem Cloning for dropping the objects that cannot be processed by Db2 CATMAINT utility, in order to re-create them with new schemas. These objects include views, materialized query tables, SQL scalar functions, triggers, and native SQL procedures. The data set pointed to by DROP-DDL-DDN must have an LRECL of 80 and RECFM of FB.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: DRDDN

**JOURNAL-DSN (*data set name*)
or JOURNAL-DDN (*ddname*)**

Supplies either the data set name of the Db2 Cloning Tool journal file or the ddname of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

SCHEMA-MASKS (*mask pairs*)

SCHEMA-MASKS are specified in *oldvalue newvalue* pairs. SCHEMA-MASKS are processed in order. The first hit of an old value mask to a source schema value is the new value mask of that pair that is used to create the target schema value. Only schemas with values no longer than 8 symbols will be translated.

- Default: None
- Required: Yes
- Restrictions: Schema mask values must not be longer than 8 symbols. For additional requirements, see the topic [“SCHEMA-MASKS Considerations”](#) on page 436.
- Short form: SM

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2SQL for this Db2 subsystem or data sharing group.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME. *name* can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

EXECUTE-CREATE-DDL (**YES | NO)**

This parameter allows you to specify whether the DDL that is generated in the CREATE-DDL-DDN DD is run. Specify NO if a different DDL job should be run, or if the dropped objects are not needed after the cloning.

- Default: YES
- Required: No
- Restrictions: None
- Short form: XCRDDL

RESUME

Specifies that the DB2SCHEMA-UPDATE step is being run a second time using the same parameters as the last time. This parameter can be used when a previous run finished with an error. When the error occurred during schema searching or DDL generation, the entire process will resume from the beginning. When the error occurred during DDL execution, the new run starts from the DDL statement that failed. When the error occurred during Db2 CATMAINT execution, the new run starts from the invocation of Db2 CATMAINT. If the previous run was not a simulation, and input parameters have changed, the job finishes with an error.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with SIMULATE.

SIMULATE

Specifies that the generated SQL commands will not be issued and catalog update will not be performed.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with RESUME.
- Short form: SIM

USE-DDL-SQLID (*sqlid*)

This parameter is used to specify the SQLID to be used in the CURRENT SQLID parameter of DDL statements for the DB2SCHEMA-UPDATE command. The SQLID that is provided can be up to eight characters long.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SQLIDDDL

DB2SCHEMA-UPDATE step JCL example

This topic contains an example of DB2SCHEMA-UPDATE step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDUPDS.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2SCHEMA-UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2SUPD' , CLASS=A , MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010 , REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD , DISP=SHR
4 // DD DSN=DSNxxx .SDSNLOAD , DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) , DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //JOURNAL DD DSN=CKZ.JRNL , DISP=OLD
9 //DRDDN DD DSN=CKZ.DRDDL
10 //CRDDN DD DSN=CKZ.CRDDL
11 //CKZIN DD *
12 DB2SCHEMA-UPDATE -
13 DB2-SSID(DB1T) -
14 DROP-DDL-DDN(DRDDN) -
15 CREATE-DDL-DDN(CRDDN) -
16 SCHEMA-MASKS(SRCSCHM TRGSCHM) -
17 JOURNAL-DDN(JOURNAL)
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.

3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, rather than a data set name.
7. DD for generated DROP DDL statements.
8. DD for generated CREATE DDL statements.
9. DB2-SSID parameter specifying the Db2 SSID, DB1T, that will be connected to.
10. SCHEMA-MASKS parameter specifying the pair of source and target values for the schema that need to be changed.

DB2SETLOG

This command is not required. This command is used only for online cloning of a Db2 subsystem using methods other than consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

DB2SETLOG is used to suspend or resume a source Db2 subsystem, via a Db2 SET LOG command, as part of online cloning a Db2 subsystem. For a Db2 data sharing group, the SET LOG command must be run on each active member.

For SUSPEND, a SET LOG LOGLOAD(0) command is issued followed by a SET LOG SUSPEND command.

For RESUME, a SET LOG RESUME command is issued.

The user ID that runs the DB2SETLOG command must be authorized to connect to the Db2 subsystem and issue the SET LOG commands.

DB2SETLOG command syntax

DB2SETLOG

Required keywords:

```
DB2-SSID( db2 ssid )
SUSPEND | RESUME
```

Optional keywords:

```
SIMULATE
```

DB2SETLOG command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2SETLOG

Optional command to suspend or resume a Db2 subsystem.

- Required: No
- Restrictions: This command is only used for cloning of a Db2 subsystem.

DB2-SSID(db2-ssid)

This parameter supplies the Db2 SSID of the Db2 subsystem to be suspended or resumed. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

SUSPEND

Specifies that a suspend of the Db2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with RESUME.

RESUME

Specifies that a resume of the Db2 subsystem will be done.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with RESUME.

SIMULATE

Specifies that the Db2 SET LOG command will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

DB2SETLOG step JCL example

There are two DB2SETLOG step JCL examples: one to suspend a Db2 subsystem, and the other to resume a Db2 subsystem. Sample JCL for both can be found in the installation library SCKZJCL in member CKZDSETL.

Suspend a Db2 subsystem

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2SETLOG step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2SETLOG', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=6M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 // DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //CKZIN DD *
9 DB2SETLOG -
10 DB2-SSID(DB1S) -
11 SUSPEND
12 /**
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.

4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying the Db2 ssid, DB1S, of the Db2 subsystem that will be suspended.
7. SUSPEND parameter specifying that the Db2 subsystem will be suspended.

Resume a Db2 subsystem

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2SETLOG step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'DB2SETLOG', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=6M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 // DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //CKZIN DD *
9 DB2SETLOG -
10 DB2-SSID(DB1S) -
11 RESUME
12 //*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying the Db2 ssid, DB1S, of the Db2 subsystem that will be resumed.
7. RESUME parameter specifying that the Db2 subsystem will be resumed.

DB2SQL

This command is optional and is used only for cloning of a Db2 subsystem. DB2SQL generates and runs the SQL statements necessary to update the Db2 catalog.

The VCATNAME, storage group names, and volumes are updated in the Db2 catalog. The VCATNAME and storage group names come from the Db2 Cloning Tool DB2UPDATE command from the DB2-HLQS and STOGROUPS keywords, and the volumes come from the Db2 Cloning Tool COPY command. The DB2SQL command updates the GROUP_MEMBER field in the SYSIBM.SYSDATABASE table for work databases, for each data sharing member of the data sharing group that was journaled during DB2UPDATE command execution.

For COPY-BY-DS, the volumes do not come from the COPY-BY-DS command. Because the COPY-BY-DS command uses SMS to assign volumes to data sets, and data sets are copied individually instead of at the volume level, all of the data sets that reside on the source volume are not likely to be copied to a single target volume. Also, the number of target volumes may be different than the number of source volumes. For these reasons, SYSVOLUMES is not updated with a straight 1:1 mapping of sources to targets, but with VOLID = "*", using the storage group names as above.

The DB2SQL command can be run only after the DB2UPDATE command has been run.

The user ID that runs the DB2SQL command must be defined in the special zparms as either SYSADM or SYSADM2.

To update the Db2 catalog on the target Db2 subsystem, the Db2 Cloning Tool DB2SQL command uses SQL and has a plan and package that must be installed. For more information about installing the plan and package, see the topic [“Cloning a Db2 subsystem”](#) on page 112.

The WLM ENVIRONMENT values that Db2 uses for stored procedures and functions can be updated by using the WLM-ENVIRONMENT-MASKS keyword.

When the DB2SQL command runs, a user-defined index on the Db2 catalog that takes an extent before the DB2SQL command completes can create problems. User-defined indexes on the Db2 catalog are logically part of the Db2 catalog, but they are treated by Db2 as non-system objects. If Db2 must extend a user-defined index, it uses the STOGROUP definitions in effect at that time. If the updates to the STOGROUP definitions have not been completed, Db2 might produce an error when attempting to extend the user-defined index. User-defined indexes created after Db2 APAR PI11316 has been applied should not have this problem.

For more information about this command, see [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2SQL command syntax

DBSQL

Required keywords:

```
DB2-SSID( db2 ssid )
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
DB2-HLQS( sourcealias1, targetalias1, ... )
DB2-NAME ( name )
HLQ-NOT-UPDATED( RC( nn | 4 ) )
LISTSQL( Y | N )
SIMULATE
WLM-ENVIRONMENT-MASKS( mask pairs )
WLM-ENV-NOT-UPDATED( RC( nn | 4 ) )
DATACLAS-MASKS( mask pairs )
DATACLAS-NOT-UPDATED( RC( nn | 4 ) )
MGMTCLAS-MASKS( mask pairs )
MGMTCLAS-NOT-UPDATED( RC( nn | 4 ) )
STORCLAS-MASKS( mask pairs )
STORCLAS-NOT-UPDATED( RC( nn | 4 ) )
STOGROUP-NOT-UPDATED ( RC ( nn ) )
```

WLM ENVIRONMENT and SMS class considerations

Oldvalue Syntax

The old value filter mask is used to select the WLM ENVIRONMENT or the SMS class values against which to apply the new value mask. The allowable filter characters are shown in the following table:

| <i>Table 80. Filter characters allowed for oldvalue filter masks</i> | |
|--|---|
| Character | Description |
| * | A single asterisk represents 0 to nn characters of any value. |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| < | A lesser-than sign represents one non-numeric character, national symbols included. |

| <i>Table 80. Filter characters allowed for oldvalue filter masks (continued)</i> | |
|--|---|
| Character | Description |
| > | A greater-than sign represents one numeric character. |

Example: Old value filter mask = D%2P* would match WLM ENVIRONMENT value = DB2PENV1

For information about filters and ACS masks, refer to the topic [“Use of the U.S.A. EBCDIC code set”](#) on page 25.

Newvalue Syntax

The new value filter mask is used to rename the WLM ENVIRONMENT or SMS class values selected by the old value filter mask. The allowable filter characters are shown in the following table:

| <i>Table 81. Filter characters allowed for newvalue filter masks</i> | |
|--|---|
| Character | Description |
| * | A single asterisk represents 0 to nn characters of any value. The single asterisk may only be used as the last character of the mask. |
| % | A percent sign represents one non-blank character. |

Example: Original WLM ENVIRONMENT value: DB2PWLM1 New value filter mask: TS%T* New WLM ENVIRONMENT value: TS2TENV1

For information about filters and ACS masks, refer to the topic [“Use of the U.S.A. EBCDIC code set”](#) on page 25.

DB2SQL command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2SQL

Optional command to update the Db2 catalog.

- Required: No
- Restrictions: The DB2SQL command can be run only after the DB2UPDATE command has been run.

DB2-SSID(*db2-ssid*)

This parameter supplies the Db2 SSID of the Db2 subsystem to connect to. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

Supplies either the data set name of the Db2 Cloning Tool journal file or the ddname of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-HLQS(*sourcealias1, targetalias1, ...*)

Specify the source and target Db2 data set high-level qualifiers (1-8 character aliases). This keyword is only applicable when the Db2 directory LOB is compressed. It allows you to override the value that was specified in the DB2-HLQS keyword in the prior DB2UPDATE command.

- Default: The default is the value that was specified in the DB2-HLQS keyword in the prior DB2UPDATE command.
- Required: No
- Restrictions: Only applies when the Db2 directory LOB (DSNDB01.SYSDBDXA) is compressed.
- Short form: DHLQS

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this Db2 subsystem or data sharing group.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME. Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

HLQ-NOT-UPDATED(RC(*nn* | 4))

Supplies the return code that is used if a VCAT in the Db2 directory or catalog is not updated. This keyword is only applicable when the Db2 directory LOB is compressed. It allows you to override the value that was specified in the HLQ-NOT-UPDATED keyword in the prior DB2UPDATE command. If you do not change a VCAT in the Db2 directory or catalog, the target Db2 system might access table and index spaces on the source Db2 system.

- Default: The default value is the value that was specified in the HLQ-NOT-UPDATED keyword in the prior DB2UPDATE command.
- Required: No
- Restrictions: Only applies when the Db2 directory LOB (DSNDB01.SYSDBDXA) is compressed.

LISTSQL(Y | N)

Specifies that the generated SQL is to be included in the listing.

- Default: N
- Required: No
- Restrictions: None

SIMULATE

Specifies that the generated SQL commands will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

WLM-ENVIRONMENT-MASKS (*mask pairs*)

WLM-ENVIRONMENT-MASKS are specified in *oldvalue newvalue* pairs. WLM-ENVIRONMENT-MASKS are processed in order. The first hit of an old value mask to a source WLM ENVIRONMENT value is the new value mask of that pair that is used to create the target WLM ENVIRONMENT value.

- Default: None

- Required: No
- Restrictions: See the topic [“WLM ENVIRONMENT and SMS class considerations”](#) on page 443.

WLM-ENV-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a WLM ENVIRONMENT value in SYSIBM.SYSROUTINES that is not updated.

- Default: RC(4)
- Required: No
- Restrictions: Used only if WLM-ENVIRONMENT-MASKS is specified.

DATACLAS-MASKS (*mask pairs*)

DATACLAS-MASKS are specified in *oldvalue newvalue* pairs. DATACLAS-MASKS are processed in order. The first hit of an old value mask to a source DATACLAS value is the new value mask of that pair that is used to create the target DATACLAS value.

- Default: N
- Required: No
- Restrictions: See the topic [“WLM ENVIRONMENT and SMS class considerations”](#) on page 443.

DATACLAS-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a DATACLAS value in SYSIBM.SYSSTOGROUP that is not updated.

- Default: RC(4)
- Required: No
- Restrictions: Used only if DATACLAS-MASKS is specified. Only valid for Db2 Version 9.1.

MGMTCLAS-MASKS (*mask pairs*)

MGMTCLAS-MASKS are specified in *oldvalue newvalue* pairs. MGMTCLAS-MASKS are processed in order. The first hit of an old value mask to a source MGMTCLAS value is the new value mask of that pair that is used to create the target MGMTCLAS value.

- Default: None
- Required: No
- Restrictions: See the topic [“WLM ENVIRONMENT and SMS class considerations”](#) on page 443.

MGMTCLAS-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a MGMTCLAS value in SYSIBM.SYSSTOGROUP that is not updated.

- Default: RC(4)
- Required: No
- Restrictions: Used only if MGMTCLAS-MASKS is specified. Only valid for Db2 Version 9.1.

STORCLAS-MASKS (*mask pairs*)

STORCLAS-MASKS are specified in *oldvalue newvalue* pairs. STORCLAS-MASKS are processed in order. The first hit of an old value mask to a source STORCLAS value is the new value mask of that pair that is used to create the target STORCLAS value.

- Default: None
- Required: No
- Restrictions: See the topic [“WLM ENVIRONMENT and SMS class considerations”](#) on page 443.

STORCLAS-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a STORCLAS value in SYSIBM.

- Default: RC(4)
- Required: No
- Restrictions: Used only if STORCLAS-MASKS is specified. Only valid for Db2 Version 9.1.

STOGROUP-NOT-UPDATED (RC (nn))

Supplies the return code that is used when any STOGROUP could not be updated because it refers to one or more non-SMS VOLID rows in SYSVOLUMES.

- Default: None
- Required: No
- Restrictions: None

DB2SQL step JCL example

This topic contains an example of DB2SQL step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDSQL.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2SQL step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2SQL' , CLASS=A , MSGCLASS=X
//S1 EXEC PGM=CKZ00010 , REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD , DISP=SHR
3 // DD DSN=DSNxxx . SDSNLOAD , DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) , DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ. JRNL , DISP=OLD
//CKZIN DD *
DB2SQL
7 DB2-SSID(DB1T) -
8 LISTSQL(Y) -
6 JOURNAL-DDN(JOURNAL)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, rather than a data set name.
7. DB2-SSID parameter specifying Db2 ssid, DB1T, that will be connected to.
8. LISTSQL parameter specifying the generated SQL statements should be included in the listing.

DB2START

This command is not required. This command is used only for cloning of a Db2 subsystem.

DB2START is used to start a Db2 subsystem, via a z/OS START Db2 command, as part of cloning a Db2 subsystem. The Db2 subsystem can be started in a special maintenance mode, maintenance mode, or normal mode. The SPECIAL, MAINT, and NORMAL keywords are used to specify the Db2 subsystem start mode.

Using the SPECIAL keyword, the Db2 subsystem will be started in maintenance mode with a special dsnzparm module. This special dsnzparm module is checked to verify that it has the correct settings:

DEFER,ALL and the Db2 catalog updatable attribute. (See the topic [“Cloning a Db2 subsystem”](#) on page 112 for how to set up the special dsnzparm module.)

After the START Db2 command is issued, DB2START waits for the Db2 subsystem to complete its start-up. The WAIT keyword is used to specify how long this wait is and the return code to be used when the wait time is exceeded. When the LIGHT keyword is specified, DB2START will wait until the Db2 subsystem stops after it starts up.

Using the REPLY-TO-RESTART-WTOR(Y) keyword, DB2START will automatically reply Y to the Db2 DSNJ245I or DSNJ246I restart WTOR message. These WTOR messages can occur when a DB2ALTERBSDS command using the COLD-START or SLB-START keywords was run prior to DB2START.

Using the STOP-WAITING-IF-DSNR020I(Y ,RC(n)) keyword, DB2START will stop waiting if a DSNR020I WTOR is issued and will provide the return code specified in the keyword (or the default return code if not specified). The DSNR020I WTOR message can occur when cloning a data sharing group from a Db2 system level backup (Db2 BACKUP SYSTEM). When the target Db2 members are started, they may receive the DSNR020I WTOR if a conditional restart record was created in each member to have its log truncated to the BACKUP SYSTEM data complete LRSN. The conditional restart record can be created by the DB2ALTERBSDS command with the SLB-START keyword. To allow for better automation of the cloning process in this situation, use the STOP-WAITING-IF-DSNR020I(Y,RC(n)) keyword. If the start jobs are being run serially by a job scheduler, it is recommended that the DB2START of all but the last member should include the STOP-WAITING-IF-DSNR020I keyword with a value of Y. This will allow members to be started concurrently. After the last member has been started, a DB2START for the first member should be reissued using the WAITONLY and DB2-ALREADY-RUNNING(RC(0)) keywords. Completion of this DB2START with a return code of 4 or less indicates that the first member is up and ready for the remainder of the Db2 conditioning jobs. For an example of how to use the STOP-WAITING-IF-DSNR020I keyword, refer to the step for running Db2 conditioning commands (Step 8) in [“Db2 subsystem cloning from a Db2 BACKUP SYSTEM backup”](#) on page 1105.

The user ID that runs the DB2START command must be authorized to issue z/OS START Db2 commands and to connect to the Db2 subsystem. When the SPECIAL keyword is used, the user ID must also be defined in the special dsnzparm as either SYSADM or SYSADM2. When the MAINT keyword is used, the user ID must also be defined in the dsnzparm as either SYSADM or SYSADM2.

A target Db2 subsystem will use the same buffer pool specifications as its corresponding source Db2 subsystem. If the buffer pool definitions in the source Db2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target Db2 subsystem until ALTER BUFFERPOOL commands can be issued.

If the Db2 system is data sharing and the DDF LOCATION is being changed, when the target Db2 systems are started they may issue the message:

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH THE LOCATION NAME
location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way Db2 rebuilds the SCA when the target Db2 system is initially started.

If there are DDF transactions active during the cloning, when the target Db2 systems are started they may issue the message:

```
DSNL034E DDF CANNOT BE STARTED BECAUSE OF BSDS INCONSISTENCIES
```

To resolve this condition and allow DDF to start, the following Db2 command must be issued on the target Db2 system:

```
-RESET INDOUBT LUNAME(*) FORCE
```

This Db2 command should not be issued until after the DB2FIX command with the DATABASES(APPLICATION) keyword has been run.

DB2START command syntax

DB2START

Required keywords:

```
DB2-SSID( db2 ssid )  
NORMAL | MAINT | SPECIAL
```

Optional keywords:

```
DB2-ALREADY-RUNNING( RC( nn | 8 ) )  
DSNZPARAM( zparm-name )  
LIGHT  
MSTR-DETECT-WAIT( [ nnn | 1 ] [, RC( nn | 8 ) ] )  
REPLY-TO-RESTART-WTOR( Y | N )  
SIMULATE  
STOP-WAITING-IF-DB2-STOPS( N )  
| STOP-WAITING-IF-DB2-STOPS( [ Y ] [, RC( nn | 8 ) ] )  
STOP-WAITING-IF-DSNR020I( N )  
| STOP-WAITING-IF-DSNR020I( Y [, RC( nn | 3 ) ] )  
WAIT( [ nnn | 5 ] [, RC( nn | 8 ) ] )  
WAITONLY
```

DB2START command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2START

Optional command to start a Db2 subsystem.

- Required: No
- Restrictions: This command is only used for cloning of a Db2 subsystem.

DB2-SSID(*db2-ssid*)

This parameter supplies the Db2 SSID of the Db2 subsystem to start. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

NORMAL

Specifies that a normal start of the Db2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with MAINT and SPECIAL.

SPECIAL

Specifies that a special start of the Db2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NORMAL and MAINT. If SPECIAL is specified, DSNZPARAM must also be specified.

MAINT

Specifies that a MAINT start of the Db2 subsystem will be done.

- Default: None
- Required: Yes

- Restrictions: Mutually exclusive with NORMAL and SPECIAL

DB2-ALREADY-RUNNING(RC(*nn* | 8))

This parameter supplies the return code that will be used if the Db2 subsystem is already running.

- Default: RC(8)
- Required: No
- Restrictions: None

DSNZPARM(*zparm-name*)

This parameter supplies the dsnzparm module name that will be used in the start Db2 command.

- Default: None
- Required: Only required if SPECIAL is specified.
- Restrictions: None

LIGHT

Specifies that LIGHT(YES) will be used in the start command.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with STOP-WAITING-IF-DB2-STOPS.

MSTR-DETECT-WAIT([*nnn* | 1] [, RC(*rr* | 8)])

nnn specifies the maximum time in minutes that DB2START should wait for the Db2 xxxxMSTR address space to start executing. If the specified time limit expires before the Db2 xxxxMSTR address space has started executing, DB2START will terminate with a return code of *rr*.

- Default: MSTR-DETECT-WAIT(1,RC(8))
- Required: No
- Restrictions: None

REPLY-TO-RESTART-WTOR(Y | N)

Specifies whether DB2START should make a Y reply to the Db2 DSNJ245I or DSNJ246I restart WTOR message.

This option addresses the situation where Db2 puts out a WTOR during its start up and it is desired to automatically make a Y reply to it. The Db2 restart WTOR needing the reply is a result of a conditional restart record that is causing either a cold start or a truncation of the Db2 log. This type of conditional restart record can be created by the DB2ALTERBSDS command using the COLD-START or SLB-START keywords.

The full text of the Db2 restart WTOR messages addressed by this keyword:

```
DSNJ245I CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT RBA rrr.
```

```
REPLY Y TO CONTINUE, N TO CANCEL
```

```
DSNJ246I CONDITIONAL RESTART RECORD INDICATES COLD START AT RBA rrr.
```

```
REPLY Y TO CONTINUE, N TO CANCEL
```

- Default: N
- Required: No
- Restrictions: None

SIMULATE

Specifies that the start command will not be issued.

- Default: None

- Required: No
- Restrictions: None.
- Short form: SIM

STOP-WAITING-IF-DB2-STOPS(N)
or STOP-WAITING-IF-DB2-STOPS([Y] [, RC(nn | 8)])

Specifies whether DB2START should stop waiting for the Db2 subsystem to start if the Db2 subsystem stops prematurely.

N specifies that DB2START should not stop waiting if the Db2 subsystem stops prematurely. The WAIT keyword will control how long DB2START will wait and the return code that is issued when the wait time has been exceeded. Y specifies that DB2START should stop waiting if the Db2 subsystem stops prematurely and DB2START will terminate with a return code of *nn*.

- Default: STOP-WAITING-IF-DB2-STOPS (Y,RC(8))
- Required: No
- Restrictions: Mutually exclusive with LIGHT.

STOP-WAITING-IF-DSNR020I(N)
or STOP-WAITING-IF-DSNR020I(Y [, RC(nn | 3)])

Specifies whether DB2START should stop waiting for the Db2 subsystem to start if the DSNR020I WTOR is received.

N specifies that DB2START should continue to wait if the DSNR020I WTOR is received. The WAIT keyword will control how long DB2START will wait and the return code that is issued when the wait time has been exceeded. Y specifies that DB2START should stop waiting if the DSNR020I WTOR is received. DB2START will terminate with a return code of *nn*. The full text of the Db2 restart WTOR message addressed by this keyword is DSNR020I START MEMBER *member*, OR REPLY 'NO' OR 'QUIESCED'.

- Default: STOP-WAITING-IF-DSNR020I(N)
- Required: No
- Restrictions: None

WAIT([nnn | 5] [,RC(rr | 8)])

nnn specifies the maximum time in minutes that DB2START should wait for the Db2 subsystem to start. If the specified time limit expires before the Db2 subsystem has started, DB2START will terminate with a return code of *rr*.

- Default: WAIT(5,RC(8))
- Required: No
- Restrictions: None

WAITONLY

Specifies that the start command will not be issued but the wait for the Db2 subsystem to start will be done.

- Default: None
- Required: No
- Restrictions: None

DB2START step JCL example

This topic contains an example of DB2START step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2START step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1  //??????? JOB , 'DB2START' ,CLASS=A,MSGCLASS=X
2  //S1 EXEC PGM=CKZ00010,REGION=6M
3  //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
4  // DD DSN=DSNxxx.SDSNEXIT,DISP=SHR
5  // DD DSN=DSNxxx.SDSNLOAD,DISP=SHR
6  //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
7  //CKZPRINT DD SYSOUT=*
8  //SYSUDUMP DD SYSOUT=*
9  //CKZIN DD *
   DB2START -
7   DB2-SSID(DB1T) -
8   SPECIAL -
9   DSNZPARM(DB1TSPEC)
   /**
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 EXIT library, must be authorized.
4. Db2 LOAD library must be authorized.
5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
6. DD for CKZPRINT output.
7. DB2-SSID parameter specifying the Db2 ssid, DB1T, of the Db2 subsystem that will be started.
8. SPECIAL parameter specifying that the Db2 subsystem will be started in maintenance mode with the special dsnzparm DB1TSPEC.
9. DSNZPARM parameter specifying the module name of the special dsnzparm.

DB2STOP

This command is not required. This command is used only for cloning of a Db2 subsystem.

DB2STOP is used to stop a Db2 subsystem, via the Db2 STOP Db2 command, as part of cloning a Db2 subsystem.

After the STOP Db2 command is issued, DB2STOP waits for the Db2 subsystem to terminate. The WAIT keyword is used to specify how long this wait is and the return code to be used when the wait time is exceeded.

The user ID that runs the DB2STOP command must be authorized to connect to the Db2 subsystem and issue the STOP Db2 command. When the Db2 subsystem is running in maintenance mode, the user ID must be defined in the dsnzparm as either SYSADM or SYSADM2.

DB2STOP command syntax

DB2STOP

Required keywords:

DB2-SSID(*db2 ssid*)

Optional keywords:

CASTOUT(NO | YES)
DB2-ALREADY-STOPPED(RC(*nn* | 8))
MODE(FORCE | QUIESCE)
SIMULATE

```
WAIT( [ nnn | 5 ] [ ,RC( nn | 8 ) ] )  
WAITONLY
```

DB2STOP command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2STOP

Optional command to stop a Db2 subsystem.

- Required: No
- Restrictions: This command is only used for cloning of a Db2 subsystem.

DB2-SSID(*db2-ssid*)

This parameter supplies the Db2 SSID of the Db2 subsystem to stop. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

CASTOUT(NO | YES)

This parameter supplies the CASTOUT value that will be used in the stop Db2 command.

- Default: YES
- Required: No
- Restrictions: None

DB2-ALREADY-STOPPED(RC(*nn* | 8))

This parameter supplies the return code that will be used if the Db2 subsystem is already stopped.

- Default: RC(8)
- Required: No
- Restrictions: None

MODE(FORCE | QUIESCE)

This parameter supplies the MODE value that will be used in the stop Db2 command. See the Db2 Command Reference for the meanings of the FORCE and QUIESCE keywords and their values.

- Default: QUIESCE
- Required: No
- Restrictions: None

SIMULATE

Specifies that the stop command will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

WAIT([*nnn* | 5] [,RC(*rr* | 8)])

nnn specifies the maximum time in minutes that DB2STOP should wait for the Db2 subsystem to stop. If the specified time limit expires before the Db2 subsystem has stopped, DB2STOP will terminate with a return code of *rr*

- Default: WAIT(5,RC(8))
- Required: No
- Restrictions: None

WAITONLY

Specifies that the stop command will not be issued but the wait for the Db2 subsystem to stop will be done.

- Default: None
- Required: No
- Restrictions: None

DB2STOP step JCL example

This topic contains an example of DB2STOP step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTO.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2STOP step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2STOP', CLASS=A, MSGCLASS=X
//S1 EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 // DD DSN=DSNxxx.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CKZIN DD *
DB2STOP -
6 DB2-SSID(DB1T)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. Db2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying the Db2 ssid, DB1T, of the Db2 subsystem that will be stopped.

DB2UPDATE

This command is optional. If a Db2 subsystem is being cloned, with the intent of a second Db2 subsystem accessing the renamed data sets, DB2UPDATE might be used to make some of the changes within Db2 to reflect the renamed data sets.

DB2UPDATE updates the Db2 directory and the Db2 bootstrap data sets (BSDSs). DB2UPDATE removes all SLB entries in the cloned BSDS. For data sharing, DB2UPDATE can deallocate the target Db2 XCF structures and remove the target Db2 XCF member information.

Active log data sets in the BSDSs are modified based on the rename masks supplied in the RENAME command or the COPY-BY-DS command. ARCHIVE data sets are modified only if the optional ARCHIVE keyword is specified.

The Db2 directory information that is updated resides in LOB table space SYSDBDXA in database DSND01. For Db2 10 CM and earlier, this information was in table space DBD01 in database DSND01. In the DB2UPDATE JCL, a DD statement that is named DBD01 points to the table space DBD01 in database DSND01. For Db2 10 NFM and later, DB2UPDATE detects that the information to be updated is no longer in the table space DBD01, dynamically allocates LOB table space SYSDBDXA (based on the data set name that is allocated to the DBD01 DD statement), and updates it. Another option for Db2 10 NFM and later is to add to the DB2UPDATE JCL a DD statement with the name of SYSDBDXA that points to LOB table space SYSDBDXA in database DSND01, and remove the DBD01 DD statement.

Note: If the archive logs are on tape, you can run the Db2 Cloning Tool DB2ALTERBSDS command to delete the source archive log file names from the target BSDS data sets. This optional command can run any time after Db2 Cloning Tool DB2UPDATE completes and before or after the target Db2 subsystem is up. You can leave the original names in the target BSDSs if you prefer.

For cloning using the COPY command, the target Db2 subsystem cannot be started until the Db2 Cloning Tool COPY, RENAME, and DB2UPDATE commands complete. For cloning using the COPY-BY-DS command, target Db2 subsystem cannot be started until the COPY-BY-DS and DB2UPDATE commands complete. In either case, the target Db2 subsystem must be stopped before the source subsystem is cloned again (next Db2 Cloning Tool COPY or COPY-BY-DS command).

DB2UPDATE can be run only when the previous RENAME or COPY-BY-DS was not a SIMULATE.

For data sharing, DB2UPDATE must be run for each member of the data sharing group. The first DB2UPDATE would be run for the primary member and would update the Db2 directory and that member's BSDSs. Then, DB2UPDATE would be run for each of the other members in the data sharing group and would include the BSDSONLY keyword, which would update only that member's BSDSs.

If the Db2 system is data sharing and the DDF LOCATION is being changed, when the target Db2 systems are started, they might issue the message

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH THE LOCATION NAME
location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way Db2 rebuilds the SCA when the target Db2 system is initially started.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

For more information about offline or online Db2 cloning, see the topic [Chapter 9, “Cloning Db2 subsystems,”](#) on page 125.

DB2UPDATE command syntax

DB2UPDATE

Required keywords:

```
DB2-HLQS( sourcealias1, targetalias1, ... )
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Required keywords when the source Db2 is data sharing:

```
DB2-GROUP( sourcegroup, targetgroup )
DB2-MEMBERS( sourcemember1, targetmember1, ... )
```

Optional keywords:

```
ARCHIVE
BSDSONLY | DBD01ONLY
DB2-NAME( name )
DB2-XCFCLEAN( Y | N )
DDF(
  ALIAS( alias1, alias2, ... | NOALIAS )
  DYNAMIC-ALIAS( alias1, alias2, ... | NOALIAS )
  GENERIC( genericluname | NGENERIC )
  GRPIP4( ip address | NGRPIP4 )
  GRPIP6( ip address | NGRPIP6 )
  IPNAME( ipname | NOIPNAME )
  IPV4( ip address | NOIPV4 )
  IPV6( ip address ) | NOIPV6 )
  LOCATION( location )
  LUNAME( luname | NOLUNAME )
  PASSWORD( password | NOPASSWD )
```

```

PORT( port )
RESPORT( resport )
SECPORT( secport )
)
DDF-NOT-UPDATED( RC( nn | 4 ) )
HLQ-NOT-UPDATED( RC( nn | 4 ) )
SIMULATE
STOGROUPS( mask pairs )
SYSVALUE-DDN( ddname )

```

STOGROUPS keyword considerations

Masks that use filter characters can be specified in the DB2UPDATE STOGROUPS keyword to select source stogroups and to rename the target stogroups.

Oldvalue Syntax

The old value filter mask is used to select the source stogroup name values against which to apply the new value mask. The allowable filter characters are shown in the following table:

| Character | Description |
|-----------|---|
| * | A single asterisk represents 0 to nn characters of any value. |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| < | A lesser-than sign represents one non-numeric character, national symbols included. |
| > | A greater-than sign represents one numeric character. |

Example: Old value filter mask = SRC* would match stogroup name = SRCSG001.

For information about filters and ACS masks, refer to the topic [“Use of the U.S.A. EBCDIC code set”](#) on page 25.

Newvalue Syntax

The new value filter mask is used to rename the source stogroup name value that is selected by the old value filter mask. The allowable filter characters are shown in the following table:

| Character | Description |
|-----------|---|
| * | A single asterisk represents 0 to nn characters of any value. The single asterisk may only be used as the last character of the mask. |
| % | A percent sign represents one non-blank character. |

Example: Source stogroup name = SRCSG001; new value filter mask = TGT*; new stogroup name = TGTSG001.

For information about filters and ACS masks, refer to the topic [“Use of the U.S.A. EBCDIC code set”](#) on page 25.

DB2UPDATE command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2UPDATE

Optional command to update Db2 to reflect renamed data sets.

- Required: No
- Restrictions: DB2UPDATE can be run only when the previous RENAME was not a SIMULATE.

DB2-HLQS (*sourcealias1, targetalias1, ...*)

Specify the source and target Db2 data set high-level qualifiers. (1-8 character aliases).

- Default: None
- Required: Yes
- Restrictions: None
- Short form: DHLQS

DB2-GROUP (*sourcegroup, targetgroup*)

Specify the source and target Db2 group names.

- Default: None
- Required: This keyword is required when the source Db2 is data sharing.
- Restrictions: None
- Short form: DGRP

DB2-MEMBERS (*sourcemember1, targetmember1, ...*)

Specify the source and target Db2 member names. Each of the member names must be unique.

- Default: None
- Required: This keyword is required when the source Db2 is data sharing.
- Restrictions: None
- Short form: DMEMS

JOURNAL-DSN (*data set name*)**or JOURNAL-DDN (*ddname*)**

This parameter supplies either a data set name of the Db2 Cloning Tool journal file or a ddname assumed via the JCL to point at a journal data set. The journal data set for the RENAME step must be the same data set specified for the COPY step.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

In case restarts or reruns need journal information, do not delete the data set in the last step. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

ARCHIVE

If the Db2 archive logs are on the source DASD volumes, and are copied to the target volumes, this keyword changes the data set names and VOLSERs of the Db2 archive logs in the BSDSs to the target values.

- Default: None
- Required: No
- Restrictions: None

BSDSONLY

Indicates that only the Db2 member's BSDS will be updated. No access will be attempted to the Db2 directory table space DBD01.

- Default: None
- Required: No

- Restrictions: Mutually exclusive with DBD01ONLY. If BSDSONLY is specified, DB2-GROUP and DB2-MEMBERS must also be specified.

DBD01ONLY

Indicates that only the Db2 directory table space DBD01 will be updated. No access will be attempted to the Db2 subsystem's BSDS.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with BSDSONLY.

DB2-NAME(*name*)

Specifies an arbitrary name that identifies this Db2 subsystem or data sharing group. The same name should be used for all members of a data sharing group.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME. Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

DB2-XCFCLEAN(*Y* | *N*)

Specifies that Db2 Cloning Tool should clean up the target Db2 data sharing group XCF structures and group members. If not specified in the control statements, the default is obtained from the CKZINI member.

- Default: Y
- Required: No
- Restrictions: Used only if DB2-GROUP is specified.
- Short form: DXCFCLN

DDF(

ALIAS(*alias1*, *alias2*, ... | NOALIAS)
DYNAMIC-ALIAS(*alias1*, *alias2*, ... | NOALIAS)
GENERIC(*genericluname* | NGENERIC)
GRPIP4(*ip address* | NGRPIP4)
GRPIP6(*ip address* | NGRPIP6)
IPNAME(*ipname* | NOIPNAME)
IPV4(*ip address* | NOIPV4)
IPV6(*ip address* | NOIPV6)
LOCATION(*location*)
LUNAME(*luname* | NOLUNAME)
PASSWORD(*password* | NOPASSWD)
PORT(*port*)
RESPORT(*resport*)
SECPORT(*secpport*)
)

Indicates that DB2UPDATE should modify the BSDS LBSC record. The record must exist on the source system; Db2 Cloning Tool does not add a new record.

ALIAS specifies 1–8 alias names. An alias name is of the form: *location*[:*port*][:*secpport*] where *location* is 1–16 alphanumeric characters (the first position must be alpha), and *port* and *secpport* are 1–5 numeric characters. NOALIAS nullifies the field.

DYNAMIC-ALIAS specifies 1–40 alias names. An alias name is of the form: *location*[:*port*][:*secpport*] where *location* is 1–16 alphanumeric characters (the first position must be alpha), and *port* and *secpport* are 1–5 numeric characters. NOALIAS nullifies the field. DB2UPDATE modifies the BSDS DYA record if it exists, or adds the BSDS DYA record if it does not exist. If the DYA record exists, the first *n* records for dynamic aliases are replaced with the *n* values that are specified in the DYNAMIC-ALIAS parameter. The rest of the records are nullified. If the BSDS has a DYA record and the DYNAMIC-ALIAS keyword is not specified, the record is not updated.

GENERIC is 1–8 alpha-numeric-national characters; first position must be alpha or national. NGENERIC nullifies the field.

GRPIPv4 is a dotted decimal form IP address. NGRPIPv4 nullifies the field.

GRPIPv6 is a colon hexadecimal form IP address. NGRPIPv6 nullifies the field.

IPNAME is 1–8 alpha-numeric-national characters; first position must be alpha or national. NOIPNAME nullifies the field.

IPV4 is a dotted decimal form IP address. NOIPV4 nullifies the field.

IPV6 is a colon hexadecimal form IP address. NOIPV6 nullifies the field.

LOCATION is 1–16 alphanumeric characters; first position must be alpha.

LUNAME is 1–8 alpha-numeric-national characters; first position must be alpha or national. NOLUNAME nullifies the field.

PASSWORD is 1–8 alpha-numeric-national characters; first position must be alpha or national. NOPASSWD nullifies the field.

PORT is 1–5 numeric characters.

RESPORT is 1–5 numeric characters.

SECPORT is 1–5 numeric characters. 0 nullifies the field.

- Default: None
- Required: No
- Restrictions:
 - The LBSC record must exist in the target BSDS.
 - ALIAS is only valid for Db2 Version 8 or later. The secpport option of ALIAS is only valid for Db2 Version 9.1 or later.
 - DYNAMIC-ALIAS is valid for Db2 Version 10 or later.
 - GRPIPv4, GRPIPv6, IPNAME, IPV4, IPV6, NOLUNAME, and SECPORT are only valid for Db2 Version 9.1 or later.
 - If GRPIPv4 is specified, IPV4 must also be specified.
 - If GRPIPv6 is specified, IPV6 must also be specified.
 - If IPV4(NOIPV4) is specified, GRPIPv4(NGRPIPv4) must also be specified.
 - If IPV6(NOIPV6) is specified, GRPIPv6(NGRPIPv6) must also be specified.

DDF-NOT-UPDATED(RC(*nn* | 4))

Supplies the return code that is used if there is a DDF record in the BSDS and the DDF keyword was not specified.

- Default: RC(4)
- Required: No
- Restrictions: None

HLQ-NOT-UPDATED(RC(*nn* | 4))

Supplies the return code that is used if the VSAM catalog name in the BSDS is not updated or a VCAT in the Db2 directory table space DBD01 is not updated.

If you do not change the VSAM catalog name in the BSDS, or a VCAT in the Db2 directory table space DBD01, the target Db2 system might access table and index spaces on the source Db2 system.

- Default: RC(4)
- Required: No
- Restrictions: None

SIMULATE

Specifies that BSDS records to be changed are printed with the target values, but no modifications are made to the target BSDSs; Db2 directory records to be changed indicate the number of fields that are updated, but no modifications are made to the target Db2 directory.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

STOGROUPS(*mask pairs*)

STOGROUPS are specified in *oldvalue newvalue* pairs. STOGROUPS pairs are processed in order. The first hit of an old value mask to a source STOGROUPS value is the new value mask of that pair that is used to create the target STOGROUPS value.

- Default: None
- Required: No
- Restrictions: For Db2 Version 8 and later, in any pair, the source and the target storage group names must be the same length and a storage group name can be a maximum of 30 characters.

SYSVALUE-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the Db2 VCAT pairing (DB2-HLQS) information will be written. SYSVALUE-DDN is intended to be used in a RESTORE SYSTEM utility that uses the LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*) parameters as input to the SYSVALUEDDN *ddname*.

- Default: None
- Required: Yes
- Restrictions: None

DB2UPDATE step JCL example

Three DB2UPDATE step JCL examples are included: for a non-data sharing environment, for a data sharing environment, and for CKZDUPD2. Sample JCL for these examples can be found in the installation library SCKZJCL in members CKZDUPD and CKZDUPD2.

DB2UPDATE Step JCL Example – Non-data sharing environment

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2UPDATE ', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
//*****
//* THE DB2 target MUST BE CHANGED TO MATCH THE HIGH LEVEL QUALIFIER
//* FOR THE TARGET SYSTEM
//*****
```

```

6 //BSDS01 DD DISP=OLD,DSN=target.BSDS01
6 //BSDS02 DD DISP=OLD,DSN=target.BSDS02
7 //DBD01 DD DISP=OLD,
// DSN=target.DSNDBC.DSNDB01.DBD01.I0001.A001
8 //CKZINI DD *
9 DB2UPDATE -
5 JOURNAL-DDN(JOURNAL) -
10 DB2-HLQS(SRCDB21,TGTDB21,SRADB22,TGTDB22)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2UPDATE step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.
6. DD names 'BSDS01' and 'BSDS02' pointing to names of target BSDSs.
7. DD name 'DBD01' pointing to name of target 'DSNDBC.DSNDB01.DBD01' data set.
8. CKZINI DD control statement input.
9. Db2 Cloning Tool DB2UPDATE command. This command may be specified in the same execution with COPY and RENAME if desired. When multiple commands are specified in a single execution, if any command completes with a return code GE 8, the remaining commands are flushed without executing.
10. DB2-HLQS parameter specifying that the source and target high level qualifier pairs of SRCDB21,TGTDB21 and SRCDB22,TGTDB22.

DB2UPDATE step JCL example – Data-sharing environment

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//??????? JOB , 'DB2UPDATE', CLASS=A, MSGCLASS=X
1 //S1 EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
//*****
//* THE DB2 target MUST BE CHANGED TO MATCH THE HIGH LEVEL QUALIFIER
//* FOR THE TARGET SYSTEM
//*****
6 //BSDS01 DD DISP=OLD,DSN=target.BSDS01
6 //BSDS02 DD DISP=OLD,DSN=target.BSDS02
7 //DBD01 DD DISP=OLD,
// DSN=target.DSNDBC.DSNDB01.DBD01.I0001.A001
8 //CKZINI DD *
9 DB2UPDATE -
5 JOURNAL-DDN(JOURNAL) -
10 DB2-HLQS(SRCDB21,TGTDB21,SRADB22,TGTDB22) -
11 DB2-GROUP(DBGS,DBGT) -
12 DB2-MEMBERS(DB1S,DB1T,
DB2S,DB2T) -
DDF(LUNAME(DB1TLU) PORT(1111))
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2UPDATE step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.
6. DD names 'BDS01' and 'BDS02' pointing to names of primary target member's BSDSs.
7. DD name 'DBD01' pointing to name of target 'DSNDBC.DSNDB01.DBD01' data set.
8. CKZIN DD control statement input.
9. Db2 Cloning Tool DB2UPDATE command. This command may be specified in the same execution with COPY and RENAME if desired. When multiple commands are specified in a single execution, if any command completes with a return code GE 8, the remaining commands are flushed without executing.
10. DB2-HLQS parameter specifying the source and target high level qualifier pairs of SRCDB21,TGTDB21 and SRCDB22,TGTDB22.
11. DB2-GROUP parameter specifying the source and target Db2 group names.
12. DB2-MEMBERS parameter specifying the source and target Db2 member names of DB1S, DB1T and DB2S, DB2T.

DB2UPDATE step JCL example – For CKZDUPD2

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1  //??????? JOB , 'DB2UPDATE2', CLASS=A, MSGCLASS=X
2  //S1      EXEC PGM=CKZ00010, REGION=6M
3  //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4  //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5  //CKZPRINT DD SYSOUT=*
6  //SYSUDUMP DD SYSOUT=*
7  //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
8  //*****
9  /* THE DB2 target2 MUST BE CHANGED TO MATCH THE HIGH LEVEL QUALIFIER
10 /* FOR THE TARGET SYSTEM
11 //*****
12 //BDS01   DD DISP=OLD, DSN=target2.BDS01
13 //BDS02   DD DISP=OLD, DSN=target2.BDS02
14 //CKZIN   DD *
15 DB2UPDATE -
16     JOURNAL - DDN(JOURNAL) -
17     DB2 - HLQS(SRCDB21, TGTDB21, SRCDB22, TGTDB22) -
18     DB2 - GROUP(DBGS, DBGT) -
19     DB2 - MEMBERS(DB1S, DB1T, -
20                 DB2S, DB2T) -
21     DDF(LUNAME(DB2TLU) PORT(22222) -
22     BSDSONLY
23 /*

```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.

3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2UPDATE step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.
6. DD names 'BSDS01' and 'BSDS02' pointing to names of secondary target member's BSDSs.
7. CKZIN DD control statement input.
8. Db2 Cloning Tool DB2UPDATE command. This command may be specified in the same execution with COPY and RENAME if desired. When multiple commands are specified in a single execution, if any command completes with a return code GE 8, the remaining commands are flushed without executing.
9. DB2-HLQS parameter specifying the source and target high level qualifier pairs of SRCDB21,TGTDB21 and SRCDB22,TGTDB22.
10. DB2-GROUP parameter specifying the source and target Db2 group names.
11. DB2-MEMBERS parameter specifying the source and target Db2 member names of DB1S, DB1T and DB2S, DB2T.
12. BSDSONLY parameter indicating only this member's BSDS data sets should be updated.

DB2UTILXCLEAN

This command is optional and is only used for the cloning of a Db2 system when you want to remove utility information from the target Db2 subsystem.

DB2UTILXCLEAN removes all entries from SYSUTILX. This command should be specified when utilities might be running or registered in SYSUTILX when the source Db2 subsystem is cloned. If SYSUTILX is not cleaned out, the source Db2 subsystem can become corrupted when the target Db2 subsystem cleans up the entries in SYSUTILX with a **-TERM UTIL** command. The DB2UTILXCLEAN command cleans out the target SYSUTILX and its indexes. This command can be run before starting the cloned subsystem with normal ZPARMs after cloning. If data sharing members are being removed, then DB2UTILXCLEAN can be run after DB2XCFCLEAN.

For more information on this command, see the following topics:

- [“Db2 offline cloning” on page 126](#)
- [“Db2 online cloning” on page 134](#)

If utilities were running or registered in SYSUTILX when the source Db2 subsystem was cloned, target objects might be in UT status after the cloning. The objects in UT status can be identified and **-STA DB(xxx) SP(yyy) ACCESS(FORCE)** commands can be issued against them to remove the UT status; however, this might cause the objects to be in an inconsistent state.

The CKZDUTST member of SCKZJCL can be used to run a REXX exec that identifies the table and index spaces in UT status and generates **-STA DB(xxx) SP(yyy) ACCESS(FORCE)** commands for the objects. These commands can be run to remove the UT status. The valid parameter options are ALL and UTX. ALL includes table and index spaces in UT* status, and UTX includes only table and index spaces in UTRO, UTUT, and UTRW status.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2UTILXCLEAN command syntax

DB2UTILXCLEAN

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

DB2-NAME(*name*)
SIMULATE

DB2UTILXCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2UTILXCLEAN

Optional command to clean out Db2 SYSUTILX as part of the cloning.

- Required: No
- Restrictions: Used only for the cloning of a Db2 system when you want to remove utility information from the target Db2 subsystem.

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the Db2 Cloning Tool journal file.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this Db2 subsystem or data sharing group.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME. *Name* can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that the actions to clean out Db2 SYSUTILX are shown, but no modifications are made to SYSUTILX.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2UTILXCLEAN step JCL example

This topic contains an example of DB2UTILXCLEAN step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDUTCL.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2UTILXCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to the notes that contain further information, following the sample JCL.

```
1 //??????? JOB , 'DB2UTILXCLEAN', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 // DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
9 //CKZIN DD *
10 DB2UTILXCLEAN -
11 JOURNAL-DDN(JOURNAL)
12 //*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set that is used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps and logs information that is derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool setups are used for different groups of volumes, ensure that you specify the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, instead of a data set name.

Step JCL example to identify objects with UT status

This topic contains an example of the step JCL for identifying objects with UT status. Sample JCL can be found in the installation library SCKZJCL in member CKZDUTST.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The step JCL for identifying objects with UT stats is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to the notes that contain further information, following the sample JCL.

```
1 //??????? JOB , 'CKZ CKZDUTST', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 DEL HLQ?.WRK.UTSTCMDS
```

```

      SET MAXCC=0
1  //S2      EXEC PGM=IKJEFT01,REGION=6M,PARM='CKZDUTST DB2T,ALL '
2  //STEPLIB DD DISP=SHR,DSN=DSNXXX?.SDSNLOAD
3  //SYSEXEC DD DISP=SHR,DSN=HLQ?.SCKZPARM
   //SYSTSIN DD DUMMY
   //SYSTSPRT DD SYSOUT=*
4  //CMDOUT  DD DSN=HLQ?.WRK.UTSTCMDS,
   //          UNIT=SYSDA,DISP=(,CATLG),
   //          DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
   //          SPACE=(TRK,(1,1))
   // *
   // * IF IT IS DESIRED TO EXECUTE THE GENERATED COMMANDS THE
   // * FOLLOWING STEP CAN BE UNCOMMENTED AND USED.
   // *
   // *IF1 IF (S2.RC LE 4) THEN
5  // *S3      EXEC PGM=IKJEFT01,REGION=6M
   // *STEPLIB DD DISP=SHR,DSN=DSNXXX?.SDSNLOAD
   // *SYSTSPRT DD SYSOUT=*
   // *SYSTSIN DD DISP=SHR,DSN=HLQ?.WRK.UTSTCMDS
   // *
   // *IF1 ENDIF
   // *
   //

```

1. Execution of CKZDUTST REXX exec. DB2T is the target Db2 system to use. ALL is an optional parameter that selects all objects with UT* status. To select only objects with UTRO, UTUT, or UTRW status, specify UTX. If omitted, ALL is the default.
2. Db2 LOAD library.
3. DD for SCKZPARM data set. The HLQ?.SCKZPARM library contains the CKZDUTST REXX exec.
4. DD for CMDOUT output. This DD will contain the generated **-STA DB(xxxxxxxx) SP(yyyyyyyy) ACCESS(FORCE)** commands.
5. This step runs the generated commands. It is commented out to allow for manual checking of the generated commands before running them.

DB2XCFCLEAN

This command is optional and only used for the cloning of a Db2 data sharing system.

DB2XCFCLEAN deallocates the target Db2 data sharing group XCF structures and removes the Db2 XCF group members.

The target Db2 group name comes from the prior DB2UPDATE command and is passed via the journal data set.

For more information about this command, see the following topics:

- [“Db2 online cloning with removal of data sharing members procedure” on page 140](#)
- [“Db2 online cloning with target becoming non-data sharing procedure” on page 146](#)

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SCHEMA-UPDATE, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2XCFCLEAN command syntax

```
DB2XCFCLEAN
```

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
DB2-NAME( name )  
SIMULATE
```

DB2XCFCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2XCFCLEAN

Optional command to clean out Db2 XCF structures and group members.

- Required: No
- Restrictions: Used only for the cloning of a Db2 data sharing system.

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either a data set name of the Db2 Cloning Tool journal file or a ddname assumed via the JCL to point at a journal data set. The journal data set for the RENAME step must be the same data set specified for the COPY step.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

In case restarts or reruns need journal information, do not delete the data set in the last step. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this Db2 subsystem or data sharing group.

Note: Db2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this Db2 subsystem or data sharing group.

DB2-NAME is intended to be used when more than one Db2 subsystem or data sharing group is being cloned from the same Db2 Cloning Tool COPY and RENAME. Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that which Db2 XCF structures and group members need cleaning is printed, but no modifications are made.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2XCFCLEAN step JCL example

This topic contains an example of DB2XCFCLEAN step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDXCFC.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The DB2XCFCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2XCFCLEAN' , CLASS=A , MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010 , REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD , DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) , DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL , DISP=OLD
//CKZIN DD *
DB2XCFCLEAN -
JOURNAL-DDN(JOURNAL)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2XCFCLEAN step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.

FINDUCATS

This command is not required. FINDUCATS will identify catalogs involved with source volume data sets. It does not negate the need to specify source and target catalog pairs in the COPY command.

Run FINDUCATS, at least initially, to determine involved ICF user catalogs, and then whenever you may wish to verify that the ICF user catalogs involved with source volume data sets are as specified in the COPY command.

FINDUCATS command syntax

FINDUCATS

Required keywords:

```
FROM-STORAGEGROUP(storgrp1 [ , storgrp2 ] [ , ... ] ) |
FROM-VOLSER( volser1 | volmask [ , volser2 | volmask ] [ , ... ] )
```

Optional keywords:

```
EXCLUDE-FROM-VOLSER( volser1 [ volser2 ] [ volsernnn ] )
```

FINDUCATS command and keyword definitions

Required keywords are described first, followed by optional keywords.

FINDUCATS

Optional command to locate catalogs involved with source volume data sets.

- Required: No
- Restrictions: None

FROM-STORAGEGROUP (*storgrp1* [, *storgrp2*] [, ...]) or FROMSTORAGEGROUP

Specifies the input volumes to be scanned for involved ICF user catalogs. All volumes defined to the specified storage group(s) will be included in the scan. A warning will be generated for any volumes not found but the scan will continue

- Default: None
- Required: Yes
- Restrictions: FROM-STORAGEGROUP is mutually exclusive with FROM-VOLSER.
- Short form: FRS

FROM-VOLSER (*volser1* | *volmask* [, *volser2* | *volmask*] [, ...]) or FROMVOLSER

Specifies the input volumes to be scanned for involved ICF user catalogs.

- Default: None
- Required: Yes
- Restrictions: FROM-VOLSER is mutually exclusive with FROM-STORAGEGROUP.
- Short form: FRV

EXCLUDE-FROM-VOLSER (*volser1* [*volser2*] [*volsernnn*]) or EXCLUDEFROMVOLSER

Specifies volumes, or a volume mask for volumes to be excluded from those specified for the FROM-VOLSER parameter or the FROM-STORAGEGROUP parameter.

- Default: None
- Required: No
- Restrictions: None.
- Short form: EXCFRV

FINDUCATS step JCL example

This topic contains an example of FINDUCATS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZFUCAT.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The FINDUCATS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //????????? JOB , 'FINDUCATS', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CKZIN DD *
FINDUCATS
FROM-VOLSER(ABC*)
EXCLUDE-FROM-VOLSER(ABCTST)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for Db2 Cloning Tool output. Two reports are generated:
 - Report 1 – lists each volume serial number, the ICF user catalog names(s) involved, and the aliases or high level qualifiers on the volume.
 - Report 2 – summarizes the involved ICF user catalog names. Use the catalogs found to determine the source catalogs required in the USERCATALOGS parameter of the COPY command. For each source ICF user catalog, you will need to supply a corresponding target catalog name to satisfy the USERCATALOGS requirement for source and target catalog pairs.

The alias information from report 1 will be useful in determining the target catalog aliases that must be created.

Note: FINDUCATS invokes DCOLLECT to identify ALIAS names of the source volume data sets in order to identify the correct source ICF user catalogs. DCOLLECT controls access to the DCOLLECT function, by issuing a security (RACF) check for a facility class profile of STGADMIN.IDC.DCOLLECT. If this profile exists, then Read authority is necessary. The command will not be successful if the user is not authorized (Db2 Cloning Tool does not make the call).

JRNLUPTGRADE

This command is not required. JRNLUPTGRADE is intended for situations where a current release of the Db2 Cloning Tool RENAME, DB2UPDATE, or BCSCLEAN command needs to use a journal data set created by an older release of Db2 Cloning Tool.

Db2 Cloning Tool commands that use the journal (except for the COPY and JRNLUPTGRADE commands) can only process a journal that was created by the same release of Db2 Cloning Tool.

Sometimes a journal created by a prior Db2 Cloning Tool release needs to be used by the current release. The JRNLUPTGRADE command can be used to upgrade a journal that was created by a prior release to the current release.

There are three scenarios for using JRNLUPTGRADE:

- Scenario 1: Db2 Cloning Tool COPY was run using a prior release and you want to run the current release of the Db2 Cloning Tool RENAME command.
- Scenario 2: Db2 Cloning Tool COPY and RENAME were run using a prior release and you want to run the current release of the Db2 Cloning Tool DB2UPDATE command and other Db2 conditioning commands.
- Scenario 3: The journal was created by a prior release and you want to run the current release of the Db2 Cloning Tool BCSCLEAN command.

It is not valid to run JRNLUPTGRADE between multiple runs of the RENAME command.

After running the JRNLUPTGRADE command, you might want to rename the old and new journal data sets, so that the new journal data set has the same name as the prior old journal. This way, you will not have to change the JCL in existing Db2 Cloning Tool cloning jobs.

Scenario 1: For scenario 1, the steps are:

1. Using the previous release of Db2 Cloning Tool: Run the Db2 Cloning Tool COPY command.
2. Back up the journal, the UCATBKUP data sets, and the target volumes.
3. At a later date or at a different site: Restore the journal, the UCATBKUP data sets, and the target volumes.
4. Using the current release of Db2 Cloning Tool: Run the Db2 Cloning Tool JRNLUPTGRADE command.
5. Using the current release of Db2 Cloning Tool: Run the Db2 Cloning Tool RENAME command.

6. For Db2, using the current release of Db2 Cloning Tool: Run the Db2 Cloning Tool Db2 conditioning commands.

Scenario 2: For scenario 2, the steps are:

1. Using the previous release of Db2 Cloning Tool: Run the Db2 Cloning Tool COPY command.
2. Using the previous release of Db2 Cloning Tool: Run the Db2 Cloning Tool RENAME command.
3. Using the current release of Db2 Cloning Tool: Run the Db2 Cloning Tool JRNLUPGRADE command.
4. Using the current release of Db2 Cloning Tool: Run the Db2 Cloning Tool DB2UPDATE command. Using the current release of Db2 Cloning Tool: Run the other Db2 Cloning Tool Db2 conditioning commands.

Scenario 3 can be used when a new release of Db2 Cloning Tool is being installed and repetitive clones are being run. For the repetitive clones, run BCSCLEAN before COPY for the next iteration. If the journal records used by BCSCLEAN have changed in the new release, the new release BCSCLEAN will fail. For this scenario, run JRLNUPGRADE before BCSCLEAN.

JRNLUPGRADE command syntax

JRNLUPGRADE

Required keywords:

```
{ NEW-JOURNAL-DDN( ddname ) | NEW-JOURNAL-DSN( data set name ) }  
{ OLD-JOURNAL-DDN( ddname ) | OLD-JOURNAL-DSN( data set name ) }
```

JRNLUPGRADE command and keyword definitions

Required keywords are described first, followed by optional keywords.

JRNLUPGRADE

Optional command to upgrade a journal that was created with a prior release of Db2 Cloning Tool.

- Required: No
- Restrictions: None

NEW-JOURNAL-DDN (ddname)

or NEW-JOURNAL-DSN (data set name)

This parameter supplies either the data set name of the new Db2 Cloning Tool journal file, or the DD name of the DD statement in the JCL that points to the new Db2 Cloning Tool journal file.

- Default: None
- Required: Yes
- Restrictions: None.

OLD-JOURNAL-DDN (ddname)

or OLD-JOURNAL-DSN (data set name)

This parameter supplies either the data set name of the old Db2 Cloning Tool journal file, or the DD name of the DD statement in the JCL that points to the old Db2 Cloning Tool journal file.

- Default: None
- Required: Yes
- Restrictions: None.

JRNLUPTGRADE step JCL example

This topic contains an example of JRNLUPTGRADE step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZJRNUP.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The JRNLUPTGRADE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //????????? JOB ,CKZ JRNLUPTGRADE',CLASS=A,MSGCLASS=X
//S0 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
2 DEL CKZ.UPGRADED.JRNL
  SET MAXCC=0
3 //S1 EXEC PGM=CKZ00010,REGION=6M
4 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZ.PARM(CKZINI),DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //OLDJRNL DD DSN=CKZ.JRNL,DISP=SHR
9 //NEWJRNL DD DSN=CKZ.UPGRADED.JRNL,
// RECOGR=KS,KEYLEN=64,KEYOFF=0,
// DISP=(,CATLG),UNIT=SYSALLDA,
// LRECL=600,SPACE=(CYL,(10,10))
10 //CKZIN DD *
11 JRNLUPTGRADE -
12 OLD- JOURNAL- DDN(OLDJRNL) -
13 NEW- JOURNAL- DDN(NEWJRNL)
//*
```

1. Deletes any previously existing new journal data set, in anticipation of JRNLUPTGRADE allocating a new upgraded journal for subsequent use by Db2 Cloning Tool. Because this data set is used to pass information from one Db2 Cloning Tool step to another, do not delete the new upgraded journal data set in any steps except the JRNLUPTGRADE step.
2. Execution of Db2 Cloning Tool main program.
3. Db2 Cloning Tool SCKZLOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for Db2 Cloning Tool output.
6. The old journal data set that is to be upgraded. In the sample JCL, the control statement OLD-JOURNAL-DDN(OLDJRNL) specifies that a DD with the name OLDJRNL is being used rather than a data set name.
7. The new upgraded journal data set that is created by JRNLUPTGRADE and will be used by subsequent Db2 Cloning Tool commands. In the sample JCL, the control statement NEW-JOURNAL-DDN(NEWJRNL) specifies that a DD with the name NEWJRNL is being used rather than a data set name.

ONLINECLIP

This command is not required. ONLINECLIP is intended for situations where the process used to copy the volumes (Db2 Cloning Tool COPY was not used) does not change the internal label of the target volumes back to the target volume serial; that is, the internal label reflects the source volume serial.

For Db2 Cloning Tool RENAME to function properly, the internal label needs to be corrected to contain the target volume serial.

ONLINECLIP will do this function. It will expect to find the target volumes online. It will read the internal label of the target device. If it has the corresponding source volume serial, it will be changed to the target volume serial.

For example, if you use the TSO FCESTABL command with the ONLINETGT=YES option, the target volume starts out with a target volume label. During the TSO FCESTABL, the target volume will remain online, but the volume label from the source volume will be copied to the target volume. The UCB however, still retains the target volume name. Db2 Cloning Tool RENAME requires the target volume to retain the target volume label. The Db2 Cloning Tool ONLINECLIP command reads the journal file to identify the original source and target volume pairs, scans the UCBs for the target volume names, and will change the target label so that it matches the original target volume serial number. This works because the system 'thinks' the target VOLSER is still online; it does not know the label at that device number was changed by FlashCopy. After the ONLINECLIP command has been executed, the target volumes are in the condition expected by Db2 Cloning Tool RENAME.

ONLINECLIP command syntax

ONLINECLIP

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
WRONG-VOLSER( RC( 4 | 8 ) )
```

ONLINECLIP command and keyword definitions

Required keywords are described first, followed by optional keywords.

ONLINECLIP

Optional command used to relabel the target volume(s) when the source volume label was copied but the UCB field still points to the target volume label. This can occur when TSO FCESTABL was used.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set. If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD or SHR in subsequent steps.

- Default: None
- Required: Yes
- Restrictions: None.
- Short form(s): JDSN, or JDDN

WRONG-VOLSER(RC(4 | 8))

This option addresses the situation where the internal VOLSER of a target volume does not have the expected value. The ONLINECLIP command return code for this situation will be the specified value.

- Default: 4
- Required: No
- Restrictions: None.

ONLINECLIP step JCL example

This topic contains an example of ONLINECLIP step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZOCLIP.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The ONLINECLIP step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'ONLINECLIP', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SORTMSG DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
//CKZIN DD *
ONLINECLIP -
5 JOURNAL-DDN(JOURNAL)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set passes information between Db2 Cloning Tool steps, and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the ONLINECLIP command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) is specifying that a DD with the name JOURNAL is being used rather than a data set name.

RENAME

The RENAME command is required. The RENAME step renames and catalogs the data sets from the COPY step onto target volumes.

RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.

Only the Db2 logs and BSDSs can be renamed beyond the *hlq*. All other Db2 files as shown in the following figure expect the following:

- Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB01.**
- Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB06.**
- Db2 databases – Db2 expects a specific naming standard – *vcat.DSNDBx.dbname.pname.y0001.Annn*

RENAME command syntax

RENAME

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
{ RENAME-MASKS( mask pairs ) | RENAME-MASKS-DDN( ddname ) }
```

Optional keywords:

```
DATACLAS( data class | SOURCE ) |
DATACLAS-PAIRS(
    Sourcedataclas1, Targetdataclas1,
    Sourcedataclas2, Targetdataclas2, ...
    DEFAULT_IF_NO_MATCH,
    Targetdataclasd | SOURCE )
DRIVEACS
EXCLUDE-SRCNAME( RC( 0 | NOTRENAMED-RC ) )
EXCLUDE-SRCNAME-MASKS( masks ) | EXCLUDE-SRCNAME-MASKS-DDN( ddname )
GDG-ALL-MIGRATED( SKIP ) | GDG-ALL-MIGRATED( RETAIN, RC( 0 | 4 ) )
GDG-EMPTY( SKIP ) | GDG-EMPTY( RETAIN, RC( 0 | 4 ) )
GDG-MIGRATED( ERROR ) | GDG-MIGRATED( RETAIN, RC( 0 | 4 ) )
GDG-TAPE( ERROR ) | GDG-TAPE( RETAIN, RC( 0 | 4 ) )
ISSUE-UCAT-UNALLOCATE( NO | YES [ , LOCAL |
    SYSPLEX[ , SYSGRPNAME(*ALL | *OTHER | sysgrpname)] ] )
ISSUE-VCLOSE( NO ) | ISSUE-VCLOSE( YES | BEFORE | AFTER [ , LOCAL |
    SYSPLEX[ , SYSGRPNAME(*ALL | *OTHER | sysgrpname)] ] )
MAX-TASKS( nnn | 1 )
MGMTCLAS( mgmt class | SOURCE ) |
MGMTCLAS-PAIRS(
    Sourcegmtclas1, Targetgmtclas1,
    Sourcegmtclas2, Targetgmtclas2, ...
    DEFAULT_IF_NO_MATCH,
    Targetgmtclasd | SOURCE )
MISSINGUCAT( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )
NOTRENAMED( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )
ORPHANCATENTRY( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )
RECATALOG( Y | N )
RENAME-AUDIT-LOG( N | SMF( nnn ) )
RENAME-ERROR( ABORT ) | RENAME-ERROR( CONTINUE, RC( 0 | 4 | 8 ) )
RENAME-LIST( Y | N )
RERUN
SAFE | SPEED
SIMULATE
STORCLAS( stor class | SOURCE ) |
STORCLAS-PAIRS(
    Sourcestorclas1, Targetstorclas1,
    Sourcestorclas2, Targetstorclas2, ...
    DEFAULT_IF_NO_MATCH,
    Targetstorclasd | SOURCE )
TEMPDSN( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )
UPDATE-IAM-ASSOCIATIONS( Y | N )
VALIDATE-SMS-CLASSES( Y | N )
VOLBKUP-DDN( ddname )
```

RENAME considerations

- The work data sets and output data sets created by the RENAME job step cannot reside on target volumes.
- RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.
- Name lengths: Whether changing a qualifier to a longer new name qualifier, or using the + feature to add qualifiers, be careful that new names do not exceed 44 characters or 35 for GDG base names.
- Catalog aliases and new names: Catalog aliases must be set up to match names resulting from renaming.
- Rename collisions: Be sure rename masks cannot cause two or more old names to rename to the same new name.
- Should all data sets on all volumes be renamed? If all data sets on all volumes copied should be renamed, use the NOTRENAMED option with a return code of 8, assuming the application using the target volumes requires less than an 8 return code from RENAME.
- If non-VSAM data set aliases are used, ensure that if a rename mask matches a data set, the same mask, or other masks, will match all aliases defined for the data set.
- Only the Db2 logs and BSDSs can be renamed beyond the *hlq*. All other Db2 files expect the following:
 - Db2 directory – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB01.**
 - Db2 catalog – Db2 expects a specific naming standard: *hlq.DSNDBx.DSNDB06.**

- Db2 databases – Db2 expects a specific naming standard: *vcat.DSNDBx.dbname.pname.y0001.Annn*
- An ICF catalog can be renamed, but it will not be usable as an ICF catalog.

Oldname syntax

Use the old name filter mask to select the data sets to apply the RENAME new name mask against. The allowable filter characters are shown in the following table:

| <i>Table 84. Filter characters allowed for old name filter masks</i> | |
|--|--|
| Character | Description |
| * | A single asterisk represents exactly one DSN qualifier of any value. Example: *. Or .* or .* . * combined with valid DSN characters or % means 0 to nn characters of any value. |
| ** | A double asterisk represents 0 to nn DSN qualifiers of any value. For example: **. or **. or **. ** cannot appear with any other characters within a qualifier. Three or more adjacent * are not allowed within a qualifier. ** can be used more than once in an 'old name' mask. Example: **.abcd.** or abc.**.defg.** |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| < | A less-than sign represents one non-numeric character, national symbols included. |
| > | A greater-than sign represents one numeric character. |

For example:

Filter = **.PAYROLL*.%%%%%%%%23*.DATA

would match DSN = TLQ050.PAYROLL.CYCLE23.DATA

For information about filters and ACS masks, refer to the topic [“Filtering pattern masks”](#) on page 26.

Newname syntax

Use the new name mask to rename the data sets selected by the old name filter mask. The allowable filter characters are shown in the following table:

| <i>Table 85. Filter characters allowed for new name filter masks</i> | |
|--|---|
| Character | Description |
| * | A single asterisk represents exactly one DSN qualifier. * may not be used for a partial qualifier in a 'new name' mask. Example: aaa.*bb.** would not be valid. |

Table 85. Filter characters allowed for new name filter masks (continued)

| Character | Description |
|-----------|---|
| ** | A double asterisk represents 0 to <i>nn</i> DSN qualifiers of any value. For example: **. or **. or **. ** cannot appear with any other characters within a qualifier. Three or more adjacent * are not allowed within a qualifier. ** can be used more than once in an 'old name' mask. Example: **.abcd.** or abc.**.defg.** |
| % | A percent sign represents one non-blank character. |
| ! | An exclamation point represents one national character. @ # \$ |
| +cccccc | A plus sign followed by 1 to 8 characters means 'insert this new qualifier'. |
| - | A minus sign means 'remove this qualifier from the new name'. |

For example:

Original DSN/Mask: CKZI . LAB9 . DEM01 . SDS . ORIG

Rename To DSN/Mask: CKZI . +TEST . XX%% . - . **

New Name: CKZI . TEST . XXB9 . SDS . ORIG

For information about filters and ACS masks, refer to the topic [“Filtering pattern masks”](#) on page 26.

RENAME command and keyword definitions

Required keywords are described first, followed by optional keywords.

RENAME

Renames and catalogs data sets on target volumes.

- Required: Yes
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either a data set name of the Db2 Cloning Tool journal file or a DD name assumed via the JCL to point at a journal data set. *The journal data set for the RENAME step must be the same data set specified for the COPY step.*

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each Db2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between Db2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created and referenced as OLD in subsequent steps. In case restarts or reruns need journal information, do not delete the data set in the last step. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

RENAME-MASKS (*mask pairs*)

RENAME-MASKS are specified in 'oldname' 'newname' pairs. RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.

RENAME-MASKS can be specified so that the target data set names will be exactly the same as the source data set names.

All data sets can be renamed beyond the hlq except for:

- Db2 directory – Db2 expects a specific naming standard: hlq.DSNDBx.DSNDB01.*
- Db2 catalog – Db2 expects a specific naming standard: hlq.DSNDBx.DSNDB06.*
- Db2 databases – Db2 expects a specific naming standard: vcat.DSNDBx.dbname.psname.y0001.Annn
- Db2 BSDSs, and Db2 logs can be renamed beyond the hlq.

For example:

```
RENAME-MASKS (DB2APROD.** DB2CPROD.** -  
              DB2BPROD.** DB2DPROD.**)
```

- Default: None
- Required: Yes
- Restrictions: See topic [“RENAME considerations”](#) on page 475. Mutually exclusive with RENAME-MASKS-DDN.

RENAME-MASKS-DDN (*ddname*)

This parameter specifies a DD name that points to a file containing rename mask pairs. The pairs are the same format as in the RENAME-MASKS keyword. RENAME-MASKS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with RENAME-MASKS.
- Short form: RM-DDN

DATACLAS (*data class* | **SOURCE)**

or **DATACLAS-PAIRS (*Sourcedataclas1, Targetdataclas1, Sourcedataclas2, Targetdataclas2, ... DEFAULT_IF_NO_MATCH, Targetdataclasd* | **SOURCE**)**

DATACLAS specifies the SMS DATACLAS to be used for all renamed data sets on SMS managed volumes if the value is in quotes, or will be copied from the corresponding source volume data set if SOURCE is specified. The default is SOURCE.

DATACLAS-PAIRS specifies source/target pairs for dataclas. If a data set has the specified source dataclas, the target data set will be given the paired target dataclas.

DEFAULT_IF_NO_MATCH indicates a dataclas to be assigned to any target data set whose source dataclas was not matched by any other DATACLAS-PAIR entry.

SOURCE indicates such data sets should be assigned the dataclas used by the source data set. If you want to use an SMS dataclas, SOURCE, enter SOURCE in quotes.

- Default: For DATACLAS, the default is SOURCE.
- Required: No
- Restrictions: DATACLAS and DATACLAS-PAIRS are mutually exclusive with DRIVEACS.
- Short form(s): DC, or DCP®.

DRIVEACS

Specifies that SMS class information for renamed data sets is to be derived by 'driving' ACS routines. Note that variables supplied to ACS are: DSN, STORCLAS, DATACLAS, MGMTCLAS, ACSENVIR (RENAME), XMODE (BATCH), JOB_ACCT, STEP_ACCT, USERID, GROUP, APPLIC, SYSNAME, SYSPLEX, JOBNAME, PGM, and NAME.

The use of the DRIVEACS parameter may significantly slow down the RENAME performance.

- Default: None

- Required: No
- Restrictions: DRIVEACS is mutually exclusive with DATACLAS, DATACLAS-PAIRS, MGMTCLAS, MGMTCLAS-PAIRS, STORCLAS, and STORCLAS-PAIRS.

Note: Variables supplied to ACS do not include data set attributes because Db2 Cloning Tool catalogs target data sets concurrently with changing target volumes – as opposed to cataloging data sets only after volumes have been modified. This methodology greatly improves the performance of RENAME, at the expense of the cataloging task not being aware of data set attributes that are normally supplied when driving ACS.

If SMS treatment of target volume data sets will be special, specific SMS classes can be supplied with the DATACLAS, MGMTCLAS, and STORCLAS keywords, or you can also set up a construct that triggers the Db2 Cloning Tool job name.

EXCLUDE-SRCNAME(RC(0 | NOTRENAMED-RC))

This parameter specifies the return code that will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. 0 specifies that a return code of zero will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. NOTRENAMED-RC specifies that the RC specified in the NOTRENAMED keyword will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. Using RC(0) addresses the situation where there are known data sets on the volumes that will not be renamed and it is desired to use NOTRENAMED(RC(8)) to know if some not known data sets are on the volumes.

- Default: NOTRENAMED-RC
- Required: No
- Restrictions: Only used when EXCLUDE-SRCNAME-MASKS is specified.
- Short form: XS

EXCLUDE-SRCNAME-MASKS (*masks*)

This parameter supplies a list of source data set names or masks that will not be renamed (excluded from renaming).

Data sets that are not renamed due to this keyword will be treated as if there were no rename mask match. The disposition of these not-renamed data sets is controlled by the NOTRENAMED keyword and the return code generated is controlled by the EXCLUDE-SRCNAME keyword.

To exclude a GDG and its GDS entries, two masks should be used. One mask should be the GDG base name and the other mask should be the GDG base name plus G>>>>v>>.

For example: GDG.BASE.NAME GDG.BASE.NAME.G>>>>V>>

Using two masks of this form is necessary due the different ways the data set names are stored in the volume VTOC and in the ICF catalog.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with EXCLUDE-SRCNAME-MASKS-DDN. See topic [“RENAME considerations” on page 475](#).
- Short form: XSM

EXCLUDE-SRCNAME-MASKS-DDN(*ddname*)

This parameter specifies the DD name that points to a file containing EXCLUDE-SRCNAME-MASKS. The entries are the same format as in the EXCLUDE-SRCNAME-MASKS keyword. EXCLUDE-SRCNAME-MASKS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with EXCLUDE-SRCNAME-MASKS. See topic [“RENAME considerations” on page 475](#).
- Short form: XSM-DDN

GDG-ALL-MIGRATED(SKIP)**or GDG-ALL-MIGRATED (RETAIN, RC(0 | 4))**

This command supports DFSMSHsm, FDR, and CA-Disk. This option addresses the situation where a GDG matches a RENAME mask and all the source generations have been migrated.

The GDG entry can be skipped, or the GDS entries in the GDG base record may be RETAINED with a corresponding return code of 0 or 4. If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Note: If RETAIN is used, because the migrated generations do not exist under the new name, subsequent access to the generations will fail whether one is accessed specifically or via specification of the base name only. This option is provided to retain relativity.

Note: To avoid destroying the relativity of active generations, Db2 Cloning Tool does NOT allow removing selected generations. For data that is migrated and required on the target volumes, they must be recalled prior to the COPY.

- Default: The default is SKIP. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with SKIP.

GDG-EMPTY(SKIP)**or GDG-EMPTY (RETAIN, RC(0 | 4))**

This option addresses an empty base GDG that matches a RENAME mask.

The GDG entry can be skipped, or the new base entry can be added to the target user catalog. If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

- Default: The default is SKIP. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with SKIP.

GDG-MIGRATED(ERROR)**or GDG-MIGRATED (RETAIN, RC(0 | 4))**

This option addresses the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are migrated. The migrated generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Note: If RETAINED, because the migrated generation does not exist under the new name, subsequent access to the generation will fail whether it is accessed specifically or via specification of the base name only.

To avoid destroying the relativity of active generations, Db2 Cloning Tool does not allow removing selected generations. Retaining non-existent migrated generations may be suitable for situations such as overstated GDG limits where it is normal for older generations to be migrated and hopefully never accessed, or Log Files etc. where perhaps only the current generation is kept on primary and older migrated generations are kept as a safety factor.

- Default: The default is ERROR. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with ERROR.

GDG-TAPE(ERROR)**or GDG-TAPE(RETAIN,RC(0 | 4))**

This option addresses the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are on tape. The tape generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Note: If RETAIN is specified, accessing a target tape GDS will cause a S813 ABEND whether it is accessed specifically or via specification of the base name only.

To avoid destroying the relativity of active generations, CKZ does not allow removing selected generations.

Retaining non-existent tape generations may be suitable for situations such as overstated GDG limits where older generations may have been created on tape.

- Default: The default is ERROR. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with ERROR.

ISSUE-UCAT-UNALLOCATE(NO | YES [, LOCAL | SYSPLEX [, SYSGRPNAME(*ALL | *OTHER | sysgrpname)]])

Specifies whether a catalog unallocate command, F CATALOG, UNALLOCATE (. . .), will be issued for target catalogs after the catalog update is done as part of the volume RENAME processing.

NO specifies that the unallocate command will not be issued.

YES specifies that the unallocate command will be issued.

LOCAL – the catalog unallocate command, F CATALOG, UNALLOCATE (. . .), will be issued only on the system that RENAME is running on.

SYSPLEX – the catalog unallocate command, F CATALOG, UNALLOCATE (. . .), will be routed to systems that are defined either by the SYSGRPNAME keyword in the RENAME command or by the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If a system group, or *ALL, or *OTHER is defined by SYSGRPNAME, the unallocate command will be routed to the systems of the system group, or to all systems of the sysplex, or to all systems of the sysplex that are different from the system where the RENAME command is running on, respectively. If the SYSGRPNAME is not specified in the command, then the unallocate command will be run on all systems of the system group that is defined by the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member and on the system that RENAME is running on.

SYSGRPNAME(*ALL | *OTHER | *sysgrpname*) – the system group name, or the *ALL or *OTHER value for routing of the command:

```
F CATALOG, UNALLOCATE .
```

- Default: YES, SYSPLEX
- Required: No
- Restrictions: LOCAL and SYSPLEX are mutually exclusive with NO.

ISSUE-VCLOSE(NO)

or ISSUE-VCLOSE (YES | BEFORE | AFTER [, LOCAL | SYSPLEX[, SYSGRPNAME(*ALL | *OTHER | sysgrpname)]])

Specifies whether a catalog modify command, F CATALOG, VCLOSE (*targetvolser*), will be issued as part of the volume RENAME processing. The Catalog Address Space (CAS) caches VVDS information. The modify command requests that the VVDS information cached for the target volume be refreshed.

NO specifies that the modify command will NOT be issued.

BEFORE specifies that the modify command will be issued only before the VVDS is updated.

AFTER specifies that the modify command will be issued only after the VVDS has been updated.

YES specifies that the modify command will be issued both before the VVDS is updated and after the VVDS has been updated.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

LOCAL – the catalog modify command, F CATALOG,VCLOSE(*targetvolser*), will be issued only on the system that RENAME is running on.

SYSPLEX – the catalog modify command:

```
F CATALOG,VCLOSE(targetvolser)
```

will be issued on the local system, and the modify command will be routed to the other systems that are defined either by the SYSGRPNAME keyword in the RENAME command or by the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If a system group, or *ALL, or *OTHER is defined by SYSGRPNAME, the modify command will be routed to the systems of the system group, or to all systems of the sysplex, or to all systems of the sysplex that are different from the system where the RENAME command is running on, respectively. If the SYSGRPNAME is not specified in the command, then the modify command will be run on all systems of the system group that is defined by the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member and on the system that RENAME is running on.

SYSGRPNAME(*ALL | *OTHER | *sysgrpname*) – the system group name, or the *ALL or *OTHER value for routing of the command:

```
F CATALOG,VCLOSE.
```

- Default: YES, LOCAL
- Required: No
- Restrictions: LOCAL and SYSPLEX are mutually exclusive with NO.

MAX-TASKS (*nnn* | 1)

Specifies the maximum subtasks to be used for volume processing in the RENAME step. Increasing the number of subtasks can greatly reduce the volume processing time. Provide a reasonably large region size and increase the number of subtasks until a point of no gain is realized. Because contention issues contributing to the point of no gain will vary by installation, experiment with this parameter to determine the optimum setting. Although the RENAME step can be rerun, because volume VTOCs, VTOC indexes, and VVDSs are restored when RERUN is specified, comparative times will be distorted. Hence, when experimenting, run the process completely over from the COPY step. This exercise can also be a good time to experiment with the COPY COPYCMDLIMIT option.

Another factor to consider is the cataloging time. The cataloging is performed by a subtask of RENAME that runs in parallel with the volume processing subtasks. If the number of data sets involved is high compared to the number of volumes, because the cataloging subtask may take longer than all volume processing subtasks, increasing the volume processing MAX-TASKS may have no effect on the complete run time of RENAME. The time the cataloging process ends, relative to the end time of the entire step, can be found in the output.

The maximum value is 255.

- Default: 1
- Required: No
- Restrictions: None.

MGMTCLAS (*management class* | **SOURCE**)

or MGMTCLAS-PAIRS (*Sourcemgmtclas1, Targetmgmtclas1, Sourcemgmtclas2, Targetmgmtclas2, ... DEFAULT_IF_NO_MATCH, Targetmgmtclasd* | **SOURCE**)

MGMTCLAS specifies the SMS MGMTCLAS to be used for all renamed data sets on SMS managed volumes if the value is in quotes, or will be copied from the corresponding source volume data set if SOURCE is specified. SOURCE is the default. MGMTCLAS is mutually exclusive with DRIVEACS.

MGMTCLAS-PAIRS specifies source/target pairs for mgmtclas.

If a data set has the specified source mgmtclas, the target data set will be given the paired target mgmtclas.

DEFAULT_IF_NO_MATCH indicates a mgmtclas to be assigned to any target data set whose source mgmtclas was not matched by any other MGMTCLAS-PAIR entry.

SOURCE indicates such data sets should be assigned the mgmtclas used by the source data set. If you want to use an SMS mgmtclas, SOURCE, enter SOURCE in quotes.

- Default: The default for MGMTCLAS is SOURCE.
- Required: No
- Restrictions: MGMTCLAS and MGMTCLAS-PAIRS are mutually exclusive with DRIVEACS.
- Short form(s): MC, or MCP.

MISSINGUCAT (DELETE | KEEP [, RC (0 | 4 | 8)])

Specifies the disposition and return code to be generated for data sets found on a volume, where the data set name matches a rename mask, but the catalog back-pointer is not one of the 'source' catalogs specified in the corresponding COPY command. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member of the SCKZPARM library.

Only data sets with VVDS records have a catalog back-pointer. A return code of 8 is suggested because MISSINGUCAT errors will likely be due to the omission of a catalog name in the COPY step.

Catalog back-pointers could be in error to start with. However, to use a return code of 8 and hence avoid continual review of any errors disclosed, it is preferable to diagnose volumes so that a MISSINGUCAT detection in fact means a user catalog was omitted in the COPY step.

- Default: KEEP,RC(4)
- Required: No
- Restrictions: None.

NOTRENAMED (DELETE | KEEP [, RC (0 | 4 | 8)])

Specifies the disposition of data sets that do not match a rename mask, and the return code to be generated if at least one occurrence is detected. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member.

If you require that rename masks match all data sets on all volumes, and if they don't, you want the entire process to fail, specify a return code of 8 and the corresponding conditional use of the target volumes set for execution only if the return code is less than 8. In that the RENAME and or COPY step will be rerun, specify KEEP as the disposition.

If volumes contain data sets not needed by the application that will access target volumes, and you do not want to rename the non application data sets and want the entire process to fail if other data sets are on the volumes, specify a return code of 8 and also specify EXCLUDE-SRCNAME(RC(0)) and EXCLUDE-SRCNAME-MASKS with entries identifying all the expected non application data sets. If all the data sets on the volumes are renamed or match an entry in EXCLUDE-SRCNAME-MASKS the return code will be 0. If there is any data set on a volume that is not renamed and does not match an entry in EXCLUDE-SRCNAME-MASKS the return code will be 8 and the process will fail. Because the data sets that match an entry in EXCLUDE-SRCNAME-MASKS will not be renamed and they will also not be cataloged, especially for SMS-managed volumes, you should specify the DELETE option. This also frees up the space if allocations may occur on target volumes.

If volumes contain data sets not needed by the application that will access target volumes, and rename masks may not match all data sets, specify a return code of 0 or 4. Because not-renamed data sets will also not be cataloged, especially for SMS-managed volumes, you should specify the DELETE option. This also frees up the space if allocations may occur on target volumes.

However, note that if not all data sets are renamed, and the return code is set for this to be acceptable (0 or 4), the list of not-renamed data sets will need to be reviewed for assurance that needed data sets are indeed renamed. Obviously, it is best if the application involved "owns" the volumes and hence all data sets should be renamed. In a compromise situation, where the volumes are used by

other applications, if feasible, the advantage of renaming data sets that are not actually required, is that the NOTRENAMED return code can be set to 8 to avoid reviewing the list of not-renamed data sets each cycle.

This may be of little concern if a single, or very few rename masks will match all required data sets (i.e., you use effective naming conventions). If, for instance, all data sets belonging to the application will match the mask A1.**, it is probably a safe bet that ignoring data sets not matching the mask will not cause a problem.

Db2 Cloning Tool may be used to effectively copy only selected data sets, by not renaming all data sets. If renaming only selected data sets, with the NOTRENAMED option set to delete, the target volume will be left with only the desired data sets. Note that because the copies are by volume, track locations of target data sets will be the same as their source volume counterparts.

- Default: KEEP,RC(8)
- Required: No
- Restrictions: None.

ORPHANCATENTRY (DELETE | KEEP [, RC(0 | 4 | 8)])

Specifies the disposition and return code to be generated for data sets found in a catalog but one or more catalog volume cells are not in the list of volumes copied. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member of the SCKZPARM library.

- Default: KEEP,RC(8)
- Required: No
- Restrictions: None.

RECATALOG(Y | N)

Specifies that Db2 Cloning Tool may replace an existing catalog entry without considering it an error. If RECATALOG is not specified, and a target catalog entry for a renamed data set is found to exist, the process fails. If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Because an incorrect rename 'to' mask could accidentally replace a catalog entry for a data set not involved with the process, the setup should be established such that RECATALOG is not required. This setup can be achieved by utilizing a target catalog used only for target data sets. Because catalog entries for data sets used in each cycle of the process will be 'orphaned' by the target volumes being reused, at the beginning of the next cycle, delete and redefine the catalog, or use the Db2 Cloning Tool BCSCLEAN command to delete catalog entries from the previous cycle.

If a target catalog is populated with entries not involved with the copy process, use the BCSCLEAN command sometime between the time target volume usage ends and before the next copy process starts. By eliminating catalog entries from the previous cycle, omitting the RECATALOG option should not result in errors – unless rename 'to' masks are wrong, in which case the replacement of a catalog entry should indeed be prevented.

- Default: N
- Required: No
- Restrictions: None.

RENAME-AUDIT-LOG (N | SMF(nnn))

Specifies whether an audit log of the data sets being renamed is to be created by RENAME volume processing.

SMF(nnn) specifies that the audit log will be created and written to SMF with a record type of nnn. Valid values for nnn are 128 through 255 inclusive. SMF must be recording the specified record type. The layout of the records written can be found in member CKZRNSMF of the Db2 Cloning Tool JCL library.

- Default: N

- Required: No
- Restrictions: None.

RENAME-ERROR(ABORT)
or RENAME-ERROR(CONTINUE,RC(0 | 4 | 8))

This option specifies how processing proceeds when a RENAME error is encountered.

ABORT will terminate with an RC=8 after the first error to preserve integrity. ABORT is recommended.

CONTINUE will continue processing after most errors and the RENAME command will complete with the specified return code unless an error not handled by the CONTINUE logic is encountered.

Note: The use of CONTINUE can cause inconsistencies between the contents of the volumes and catalogs. Possible problems include:

- data sets could be cataloged but are not renamed on disk
- data sets could be renamed on disk but are not cataloged
- data sets that are not renamed on disk may not be deleted from disk
- GDG base and GDS entries will not exist in the catalog when there is a missing GDS
- a catalog entry may not point at the correct volume
- a catalog entry may be invalid
- leave uncataloged data sets on SMS managed volumes.

If this keyword is specified, Db2 Cloning Tool will not guarantee integrity and the given results will not be fixed by Db2 Cloning Tool.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

- Default: ABORT. For CONTINUE, the default is RC(8)
- Required: No
- Restrictions: RC is mutually exclusive with ABORT.

RENAME-LIST(Y | N)

Specifies whether a list of the renamed data sets is to be produced by RENAME volume processing.

- Default: N if SIMULATE is not specified. Y if SIMULATE is specified.
- Required: No
- Restrictions: None.

RERUN

Specifies that the RENAME step is being run a second time using the same target volumes resulting from the COPY step. Rerun of the RENAME step only is not possible unless the first execution specified SAFE. See the SAFE | SPEED option for more information.

SAFE causes a backup during the RENAME step of the portions of volumes changed during the volume processing – VTOC, VTOCIX, and VVDS. This backup data is used on a rerun to restore any volumes changed to the state they were in following the COPY step.

Certain errors, such as incorrect rename masks, where the list of volumes copied and the catalogs backed up are correct, can be corrected by rerunning just the RENAME step.

Examine the problem to determine if just a rerun of the RENAME step will resolve the problem or if the COPY step must be rerun. Any errors resulting from a volume being omitted or a catalog being omitted from the COPY step will require a rerun of the COPY step.

The BCSRECS and VOLBKUP data sets must not be deleted by the JCL running the RENAME RERUN.

The JCL used for RENAME RERUN is different from the JCL used for RENAME SAFE. Sample JCL for RENAME RERUN can be found in the installation SCKZJCL library in member CKZRENRR.

The contents of the BCSRECS and VOLBKUP must be from the prior run of RENAME with either the SAFE or RERUN keywords. If the BCSRECS or VOLBKUP data sets are deleted, the COPY step will need to be run again.

- Default: None.
- Required: No
- Restrictions: RERUN is mutually exclusive with SAFE and SPEED.

SAFE | SPEED

SAFE allows a rerun of the RENAME command by backing up critical volume structures that are changed during the volume processing – the VTOC, VTOCIX, and VVDS. This backup data is used on a rerun to restore any volumes changed to the state they were in following the COPY step. This adds some slight execution time for RENAME to capture the portions of target volumes modified by RENAME.

Incorrect rename masks may be a reason for needing to rerun the RENAME step. If multiple and complicated masks are required, this option is recommended. Also affecting the renaming is whether the data set naming conventions used by the application are fairly static or subject to frequent change – inferring that rename masks need to be watched.

SAFE requires the VOLBKUP parameter. Note the JCL comments that warn about making sure the VOLBKUP data set is not deleted before a rerun of RENAME. If the VOLBKUP data set is lost, the COPY step will need to be run again, provided that the opportunity for correct point-in-time images has not been lost.

If source volume access is not resumed until the entire process is complete (implying that the same point-in-time images can be re-copied), the time to rerun the COPY step may be insignificant compared to adding some overhead with the SAFE option for every cycle.

SPEED is the opposite of SAFE. The RERUN option for the RENAME step will be rejected if attempted. Correction of any errors will require the COPY and RENAME step to be run again.

- Default: SPEED.
- Required: No
- Restrictions: SAFE is mutually exclusive with SPEED and RERUN. SPEED is mutually exclusive with SAFE and RERUN.

SIMULATE

Specifies that only non-destructive RENAME activities are to be performed. RENAME SIMULATE must be used in conjunction with a COPY as the volumes and catalogs are specified only in the COPY step and are passed via the journal to the RENAME step. The COPY execution may be with or without SIMULATE.

Scenario 1 – a real COPY (without SIMULATE) followed by a RENAME SIMULATE will use only the target volumes.

Scenario 2 – a COPY SIMULATE followed by a RENAME SIMULATE will use only the source volumes because there are no target volumes yet.

If COPY DATAMOVER(PGM(NONE)) was used with SIMULATE, it can be followed by a RENAME SIMULATE.

SIMULATE will perform the masking comparisons to both catalog and volume records. Errors can be discovered such as not all data sets renamed, incomplete renaming of VSAM sphere associations and components, multivolume data sets not wholly contained in the volume list, GDG generations that stray outside of the volume list, catalog entries that match a mask but one or volumes were not 'copied', etc.

Because omitting a catalog(s) or volume(s) in the COPY step is not detected until the RENAME step, it is strongly advised that SIMULATE be used whenever significant changes are made to the involved application(s), especially if resumption of source volume access is triggered by completion of the COPY step – i.e., the opportunity for re-capturing point-in-time images has been lost.

- Default: None.
- Required: No.
- Restrictions: None.
- Short form: SIM

STORCLAS (*storage class* | **SOURCE)**

or **STORCLAS-PAIRS (*Sourcestorclas1, Targetstorclas1, Sourcestorclas2, Targetstorclas2, ...***

DEFAULT_IF_NO_MATCH, *Targetstorclasd* | **SOURCE)**

STORCLAS specifies the SMS STORCLAS to be used for all renamed data sets on SMS managed volumes if the value is in quotes, or will be copied from the corresponding source volume data set if SOURCE is specified.

STORCLAS-PAIRS specifies source/target pairs for storclas.

If a data set has the specified source storclas, the target data set will be given the paired target storclas.

DEFAULT_IF_NO_MATCH indicates a storclas to be assigned to any target data set whose source storclas was not matched by any other STORCLAS-PAIR entry.

SOURCE indicates such data sets should be assigned the storclas used by the source data set. If you want to use an SMS storclas, SOURCE, enter SOURCE in quotes.

- Default: The default for STORCLAS is SOURCE..
- Required: No.
- Restrictions: STORCLAS and STORCLAS-PAIRS are mutually exclusive with DRIVEACS.
- Short form(s): SC, or SCP

TEMPDSN (**DELETE | **KEEP [,RC(0 | 4 | 8)]**)**

Specifies the disposition of temporary data sets and the return code to be generated if at least one occurrence is discovered. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member of the SCKZPARM library.

- Default: DELETE,RC(4)
- Required: No.
- Restrictions: None.

UPDATE-IAM-ASSOCIATIONS (**Y | **N**)**

Specifies whether IAM data set associations are to be updated as part of RENAME processing. IAM must be active on the system for the updates to happen.

This option addresses the situation where there are IAM data sets that are being cloned that include AIXes and PATHs, and it is desired to update the associations to correspond with the new data set names. The association information for IAM data sets will be determined and updated by internally using IDCAMS LISTCAT and IDCAMS DEFINE RECATALOG commands.

When using RERUN, it is possible to get missing component errors when rename mask entries that cover IAM data sets have changed. This is due to the IAM association data not being in the ICF catalog, VTOC, VTOCIX, or VVDS, so Db2 Cloning Tool is unable to properly determine the changed IAM associations. When using SIMULATE, it is not possible for Db2 Cloning Tool to determine the IAM associations and verify that all the associated IAM data sets have been cloned.

- Default: N
- Required: No.
- Restrictions: None.
- Short form: UIA

VALIDATE-SMS-CLASSES (Y | N)

Specifies whether the SMS class names specified in the DATACLAS, DATACLAS-PAIRS, MGMTCLAS, MGMTCLAS-PAIRS, STORCLAS, and STORCLAS-PAIRS keywords will be validated as being defined to SMS (Y) or not (N).

This option addresses the situation where the target SMS class names are not defined on the system where RENAME is run.

- Default: Y
- Required: No.
- Restrictions: None.

VOLBKUP-DDN (*ddname*)

Specifies the DD name for the backup data set to be used for backing up target volume VTOCs, VTOCIXs, and VVDSs, to be used in the event of a rerun of the RENAME step. SAFE is required in conjunction with this parameter.

- Default: None
- Required: No.
- Restrictions: None.

RENAME step JCL example

This topic contains an example of RENAME step JCL for RENAME SAFE. Sample JCL can be found in the installation library SCKZJCL in member CKZREN.

The JCL used for RENAME RERUN is different from the JCL used for RENAME SAFE. Sample JCL for RENAME RERUN can be found in the installation library SCKZJCL in member CKZRENRR.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The RENAME step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'RENAME', CLASS=A, MSGCLASS=X
2 //S0 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 1 DEL CKZ.WRK.VOLDUMP
6 1 DEL CKZ.WRK.BCSRECS
7 2 //S1 EXEC PGM=CKZ00010, REGION=8M
8 3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
9 4 //CKZINI DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
10 5 //SORTMSG DD SYSOUT=*
11 6 //SORTWK01 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
12 //SORTWK02 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
13 //SORTWK03 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
14 //SORTWK04 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
15 //SORTWK05 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
16 //SORTWK06 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
17 7 //DRSTATS DD SYSOUT=*
18 8 //CKZPRINT DD SYSOUT=*
19 //SYSUDUMP DD SYSOUT=*
20 9 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
21 10 //BCSRECS DD DSN=CKZ.WRK.BCSRECS, UNIT=SYSALLDA, DISP=(,CATLG),
22 // SPACE=(CYL,(10,10))
23 11 //VOLDUMP DD DSN=CKZ.WRK.VOLDUMP, UNIT=SYSALLDA, DISP=(,CATLG),
24 // SPACE=(CYL,(10,10))
25 //CCKZIN DD *
26 RENAME
27 11 SAFE
28 11 VOLBKUP-DDN(VOLDUMP)
29 MAX-TASKS(5)
30 9 JOURNAL-DDN(JOURNAL)
31 NOTRENAMED(DELETE,RC(4))
32 MISSINGUCAT(DELETE,RC(8))
33 ORPHANCATENTRY(KEEP,RC(8))
34 RECATALOG(N)
35 DATACLAS(TGTDATA)
```

```

MGMTCLAS(TGTMGMT) -
STORCLAS-PAIRS(SRCSTOR1,TGTSTOR1 -
                SRCSTOR2,TGTSTOR2 -
                DEFAULT_IF_NO_MATCH,TGTSTORX) -
RENAME-MASKS( -
  ASRC.** ATGT.** -
  BSRC.** BTGT.** -
  PROD.** TEST.** -
)
/**

```

1. Deletion of volume dumps and BCSRECS in anticipation of allocating new for each execution. Because these data sets recover target volume information and remove BCS entries during a rerun of the RENAME step, DO NOT set up the rerun JCL to delete these data sets.
2. Execution of Db2 Cloning Tool main program.
3. Db2 Cloning Tool LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for sort messages.
6. DD for sort messages.
7. DD for DRSTATS, SAFE option Dump and Restore output.
8. DD for CKZPRINT output.
9. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple Db2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, rather than a data set name.
10. DD for BCSRECS. The data set pointed at by this DD is used to store data set names used if RENAME is rerun, and for the BCSCLEAN command. Required regardless of whether rerunning is anticipated or BCSCLEAN is expected to be used. The data set specified in the RENAME step must match the name used in the corresponding BCSCLEAN step. However, if multiple Db2 Cloning Tool processes are created for different applications, each must have a unique BCSRECS data set.

Note: This file cannot be striped.

11. VOLDUMP DD statement. This DD points to a data set used to back up information on target volumes, in case a rerun of the RENAME step is required. This data set is only used if the SAFE parameter is specified. The normal and abnormal disposition must be CATLG. In the sample JCL, the control statement VOLBKUP-DDN(VOLDUMP) specifies that a DD with the name VOLDUMP is being used rather than a data set name. If the number of volumes involved is high, and the VTOCs, VVDSs, and VTOC indexes are large, this data set may require a substantial amount of space.

Note: The RENAME output also includes output for each volume processed with dynamically allocated DDnames of VVOLSERX.

RESTORE-FROM-DUMPTAPES

This command is not required. RESTORE-FROM-DUMPTAPES allows Db2 Cloning Tool to use the backups on tape from a Db2 BACKUP SYSTEM DUMP utility as the source for cloning.

This command is used as part of the process to clone from Db2 BACKUP SYSTEM DUMP backups on tape. With this process, backup tapes are "restored" to the target volumes instead of the original source volumes. A detailed description of this cloning process can be found in ["Cloning scenarios"](#) on page 1091.

This command works together with the DB2GETBACKINFO command to clone directly from Db2 system level backup on tape, even when disks are not shared between the two systems. Run DB2GETBACKINFO (with the USE-DUMPTAPES keyword) against the source Db2 system to create the BACKINFO metadata

file that describes the dump tape files. Then use the backinfo file as input to RESTORE-FROM-DUMPTAPES. It is not necessary that the target volumes be online at the source Db2 system.

You can specify the target volumes directly by providing a volume list or indirectly by specifying target storage groups, as follows:

- Use only TO-VOLSER to define the target volumes. The volumes are matched against the source volumes and the dump tape volumes in the backinfo file, and assigned in the order they appear in the list.
- Use the TO-STORAGEGROUP and SOURCE-STORAGEGROUP keywords to match sources and targets one-for-one within the specified storage groups.

In either case, to prevent an attempt to restore a larger volume to a smaller one, Db2 Cloning Tool uses volume capacity as a final matching strategy.

During the restore, ADRDSSU changes the volser of the target volume to the volser of the source volume and takes the target volume offline. The volume is relabeled to the target volser, and brought back online. If you specify VARY-SCOPE(GLOBAL) (the default), Db2 Cloning Tool issues sysplex-wide VARY commands to take the volume offline before the restore and brings it back online after the restore. Use VARY-SCOPE(LOCAL) to issue the sysplex-wide VARY commands yourself.

Volume pairing process and criteria

The pairing of source and target volumes consists of two stages:

1. The first stage pairs any target volumes that have target catalogs on them if the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter is set. In this case, for each target volume that has a target catalog, a source volume with a source catalog on it is located. All target volumes with target catalogs on them must be paired at this stage; otherwise, the RESTORE-FROM-DUMPTAPES command completes with an error. If there are no target volumes with target catalogs on them, the first stage is not required.
2. In the second stage of pairing, for each source volume, a target volume is located.

In addition, KEEP-VOLUMES-SEQUENCE(Y) requires the following additional conditions:

- The source volume must be in the storage groups that are defined by the SOURCE-STORAGEGROUP keyword. The storage group position number in the SOURCE-STORAGEGROUP is called the *storage group sequence number for the source volume*.
- The target volume is defined by the TO-STORAGEGROUP keyword. The position number of the storage group in the TO-STORAGEGROUP keyword that contains the target volume is called the *storage group sequence number for the target volume*.
- The storage group sequence numbers for the source and target volumes must be equal.

First stage

All target volumes that have catalogs on them are sorted by size, in descending order. All source volumes that have catalogs on them are grouped by volume size. For each target volume, Db2 Cloning Tool attempts to match a source volume that is less than or equal to the target volume size. Source volumes that were paired with the target by using the USERCATALOGS keyword are checked first. Then Db2 Cloning Tool continues until all of the source volumes are checked.

At the end of the first stage, all of the target volumes with target catalogs on them should be paired. If not, the RESTORE-FROM-DUMPTAPES command completes with an error.

Second stage

All non-paired target volumes are grouped by volume size. All source volumes are sorted by volume size in ascending order. For each source volume, Db2 Cloning Tool attempts to find a matching target volume with a size greater or equal to the source volume size. Db2 Cloning Tool proceeds to check each target volume until all of the target volumes are checked.

At the end of the second stage, all of the source volumes should be paired. If not, the RESTORE-FROM-DUMPTAPES command completes with an error.

RESTORE-FROM-DUMPTAPES command syntax

RESTORE-FROM-DUMPTAPES

Required Keywords:

```
{ TO-VOLSER ( volser1 | volmask1...[ , volsern | volmaskn ] )  
  | TO-STORAGEGROUP( storgrp1...[ , storgrpn ] ) }
```

Optional keywords:

```
BACKINFO-DDN( ddname )  
EXCLUDE-TO-VOLSER( volser1 | volmask1...[ , volsern | volmaskn ] )  
KEEP-VOLUMES-SEQUENCE ( Y | N )  
MAX-TAPEDRIVES( nn )  
RERUN  
SIMULATE  
SOURCE-STORAGEGROUP ( sourcestorgrp1...[ , sourcestorgrpn ] )  
STATUS-DDN( ddname )  
TAPE-UNIT( name )  
TARGET-UCATS-ON-TARGET-VOLUMES( Y | N )  
USERCATALOGS( sourcecat1 targetcat1,...[ , sourcecatn targetcatn ] )  
USERCATALOGS-DDN( ddname )  
VARY-SCOPE( GLOBAL | LOCAL | SYSGRPNAME(sysgrpname) )  
VOLPAIRS-DDN( ddname )  
WAIT-TAPE-ALLOC( N | NO | nnn [,RC(rrrr | 8)] )
```

RESTORE-FROM-DUMPTAPES command and keyword definitions

Required keywords are described first, followed by optional keywords.

RESTORE-FROM-DUMPTAPES

Optional command to allow for cloning using the backup tapes from a Db2 BACKUP SYSTEM DUMP backup.

- Required: No
- Restrictions: None

TO-VOLSER (*volser1* | *volmask1*...[, *volsern* | *volmaskn*])

This parameter specifies the target volumes that are to be paired with dump volumes.

- Default: None.
- Required: No, but if not supplied, TO-STORAGEGROUP is required.
- Restrictions: Mutually exclusive with TO-STORAGEGROUP.

TO-STORAGEGROUP (*storgrp1*...[, *storgrpn*])

This parameter specifies that the target volumes that are required to pair with dump volumes are to be selected from one or more SMS storage groups. All volumes from the specified storage groups are target candidates, except volumes that are excluded via the EXCLUDE-TO-VOLSER parameter. If KEEP-VOLUMES-SEQUENCE(Y) is specified, storage groups are matched one-for-one with SOURCE-STORAGEGROUP storage groups in the order that they are specified in the parameters.

- Default: None.
- Required: No, but if not supplied, TO-VOLSER is required.
- Restrictions: Mutually exclusive with TO-VOLSER.

BACKINFO-DDN(*ddname*)

This parameter specifies the DD name that points to a file containing the backinfo data. The file named in the DD must have an LRECL of 80.

- Default: None.
- Required: No, unless the BACKINFO DD is not supplied in the JCL.

- Restrictions: None.

EXCLUDE-TO-VOLSER (volser1 | volmask1...[, volsern | volmaskn])

This parameter specifies the volumes to be excluded (not selected as targets) from either the TO-VOLSER parameter or the TO-STORAGEGROUP parameter. After target volumes are excluded, if there are fewer target volumes than dump volumes, the command fails with a return code of 8.

- Default: None.
- Required: No.
- Restrictions: None.

KEEP-VOLUMES-SEQUENCE(Y | N)

Specifies that each source volume that is defined by the SOURCE-STORAGEGROUP parameter should be restored only to the target volume defined in the TO-STORAGEGROUP parameter that has a sequence number that is equal to the source sequence number.

If KEEP-VOLUMES-SEQUENCE(Y) is set, make sure to specify the desired source and target volumes so that they have the same sequence number. When using KEEP-VOLUMES-SEQUENCE(Y) with the RESTORE-FROM-DUMPTAPES command, the sequence number is assigned as follows:

- If the SOURCE-STORAGEGROUP parameter is specified, the sequence number is the position number in which this volume is defined in the corresponding parameter, starting with 0. For example, if SOURCE-STORAGEGROUP (GR1, GR2) is specified, then all source volumes in the GR1 storage group have the sequence number 0, and all source volumes in the GR2 storage group have the sequence number 1. If no SOURCE-STORAGEGROUP parameter is used, then source volume sequence numbers are not defined.
- If the TO-STORAGEGROUP parameter is specified, then the sequence number is the position number in which this volume is defined in the corresponding parameter, starting with 0. For example, if TO-STORAGEGROUP (GR1, GR2) is defined, then all target volumes of the GR1 storage group have the sequence number 0 and all target volumes of the GR2 storage group have the sequence number 1. If a volume is defined by a TO-VOLSER parameter, then its sequence number is not defined.

If the sequence numbers do not match or are not defined, the RESTORE-FROM-DUMPTAPES command completes with an error.

- Default: N.
- Required: No.
- Restrictions: None.
- Short form: KVS

MAX-TAPEDRIVES (nn)

This parameter specifies the maximum available number of tape drives available for tape allocation (and implicitly, the number of restore subtasks). A higher value reduces elapsed run time, but may impact system resources. Valid values are 1-16, or specify an asterisk (*) to allow the RESTORE-FROM-DUMPTAPES command to set the maximum (set to 4 for this release).

- Default: 2.
- Required: No.
- Restrictions: None.

RERUN

This parameter can be used to resume processing from a previous run of the RESTORE-FROM-DUMPTAPES command that uses the same backinfo file. The contents of the status file from the previous run are used to determine the RESTORE-FROM-DUMPTAPES tasks that completed successfully, the tasks that had not started, and the tasks that were in progress. The last known state of each individual restore task, including clip and vary processing, are used to determine the steps that remain.

To use RERUN, you must:

1. Provide the STATUS DD statement on the RESTORE-FROM-DUMPTAPES command, using either a new data set with DISP=NEW, or an existing empty data set that uses the required DCB characteristics.
 2. Run the job containing the RESTORE-FROM-DUMPTAPES command. If the command completes with a return code of 0, the status file is no longer needed and can be deleted. If the return code is 8, determine and resolve the error before rerunning the job.
 3. To rerun the job, add the RERUN keyword and change the STATUS DD statement to DISP=OLD. Remove all other DD parameters on the STATUS DD except for the DSN, and remove any automatic delete processing. Then submit the job.
- Default: None.
 - Required: No.
 - Restrictions: None.

SIMULATE

This parameter verifies the syntax, determines the volumes to be processed, and displays the action that will be taken, but does not change the contents of volumes or restore the dump tapes.

- Default: None.
- Required: No.
- Restrictions: None.

SOURCE-STORAGEGROUP (*sourcestorgrp1* ... [, *sourcestorgrpn*])

This parameter specifies the source SMS storage groups. If KEEP-VOLUMES-SEQUENCE(Y) is specified, these storage groups are matched one-for-one with TO-STORAGEGROUP storage groups in the order that they are specified in the parameters. Source volumes are paired to target volumes according to these one-for-one storage group mappings; then dump volumes are paired to target volumes accordingly.

- Default: None.
- Required: No.
- Restrictions: None.

STATUS-DDN (*ddname*)

This parameter specifies the ddname of the data set that will be created on a normal (non-RERUN) run and holds the status information of the restore tasks during the job step. It must be retained after a failure (any job step that ends with return code of 8 or higher) in order to allow a RERUN of the job step. This data set will be used when RERUN is specified to determine which restores need to be done. For more information, see the RERUN parameter description.

- Default: None.
- Required: No; however, a RERUN of the job cannot be attempted unless this parameter or the STATUS DD is supplied.
- Restrictions: None.

TAPE-UNIT(*name*)

This parameter specifies the unit name to use for allocating tape devices for the dump volumes.

- Default: 3490.
- Required: No.
- Restrictions: None.

TARGET-UCATS-ON-TARGET-VOLUMES(*Y* | *N*)

Specifies that there are target ICF catalogs that reside on the target volumes, and you want the target ICF catalogs to remain on the target volumes after the RENAME step. If Y is specified, when the RENAME step completes, the target ICF catalogs will reside on target volumes.

- Default: N

- Required: No
- Restrictions: None
- Short form: TUOTV

USERCATALOGS (*sourcecat1 targetcat1, ..., [sourcecatn targetcatn]*)

This parameter specifies the source catalogs that contain data sets from source volumes, and the corresponding target catalog for renamed volume data sets. The specified source catalogs must also be in the backinfo data set in a UCAT record; otherwise, the command fails with a return code of 8.

- Default: None.
- Required: No.
- Restrictions: Can only be specified if the backinfo data set contains UCAT records.

USERCATALOGS-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the user catalog information will be written. The file named in the DD must have an LRECL of 80.

- Default: None.
- Required: No, unless USERCATALOGS is specified and the UCATS DD is not supplied in the JCL.
- Restrictions: None.

VARY-SCOPE (GLOBAL | LOCAL | SYSGRPNAME(*sysgrpname*))

When running in a sysplex, specify GLOBAL, LOCAL, or SYSGRPNAME to determine the scope of VARY commands for target volumes. If you specify GLOBAL, commands are issued on the systems of the sysplex in accordance with the value of the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If SYSPLEX_GROUPNAME_ALL is equal to *ALL or not defined, then commands are issued across the sysplex. If SYSPLEX_GROUPNAME_ALL is a system group name, then commands are issued on all systems that are defined in the system group name, and also on the system that the RESTORE-FROM-DUMPTAPES command is running on.

If you specify LOCAL, commands are issued only on the local system (the same system that the job is running on). If you specify LOCAL, resource conflicts might affect the running of the RESTORE-FROM-DUMPTAPES job, other jobs on other systems, or both.

If you specify SYSGRPNAME(*sysgrpname*), commands are issued on all systems that are defined in the system group *sysgrpname*, and also on the system that RESTORE-FROM-DUMPTAPES is running on. In this case, the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member is ignored.

- Default: GLOBAL.
- Required: No.
- Restrictions: None.

VOLPAIRS-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the volume pairs information will be written. The file named in the DD must have an LRECL of 80.

- Default: None.
- Required: No, unless the VOLPAIRS DD is not supplied in the JCL.
- Restrictions: None.

WAIT-TAPE-ALLOC(*N* | NO | *nnn* [,RC(*rrrr* | 8)])

This parameter specifies whether RESTORE-FROM-DUMPTAPES waits for each tape file to be allocated, and the return code that is issued. *nnn* specifies the maximum time in minutes that RESTORE-FROM-DUMPTAPES waits for each tape file to be allocated. If the specified time limit expires, then RESTORE-FROM-DUMPTAPES terminates with return code *rrrr*. The value of return code *rrrr* must be equal or greater than 8. If NO is specified and if tape allocation fails, RESTORE-FROM-DUMPTAPES immediately terminates with an error and the return code *rrrr*.

RESTORE-FROM-DUMPTAPES step JCL example

This topic contains an example of RESTORE-FROM-DUMPTAPES step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZRSTDT.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Two sample sets of control cards are provided in the sample JCL.

The RESTORE-FROM-DUMPTAPES step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //????????? JOB , 'CKZ RESTORE-F-DT', CLASS=A, MSGCLASS=X, NOTIFY=&SYSUID
2 //S0 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 1 DEL HLQ?.WRK.VOLPAIRS
6 2 DEL HLQ?.WRK.STATUS
7 3 DEL HLQ?.WRK.UCATS
8 SET MAXCC=0
9 4 //S1 EXEC PGM=CKZ00010, REGION=8M
10 5 //STEPLIB DD DISP=SHR, DSN=HLQ?.SCKZLOAD
11 6 //CKZINI DD DISP=SHR, DSN=HLQ?.SCKZPARM(CKZINI)
12 7 //CKZPRINT DD SYSOUT=*
13 //SYSUDUMP DD SYSOUT=*
14 //ABNLIGNR DD DUMMY DO NOT REMOVE IF USING ABENDAID
15 8 //BACKINFO DD DISP=SHR, DSN=HLQ?.WRK.BACKINFO
16 9 //VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
17 // DISP=(,CATLG), UNIT=SYSALLDA,
18 // SPACE=(CYL, (1,1))
19 10 //STATUS DD DSN=HLQ?.WRK.STATUS,
20 // DISP=(,CATLG), SPACE=(CYL, (1,1)),
21 // RECORG=KS, KEYLEN=64, KEYOFF=0,
22 // LRECL=600, UNIT=SYSALLDA
23 11 //UCATS DD DSN=HLQ?.WRK.UCATS,
24 // DISP=(,CATLG), UNIT=SYSALLDA,
25 // SPACE=(CYL, (1,1))
26 //CKZIN DD *
27 //*****
28 //* SAMPLE CONTROL STATEMENTS
29 //*****
30 //* SAMPLE 1: RESTORE FROM DUMPTAPES USING UP TO 4 TAPE DRIVES,
31 //* USING SOURCE VOLUMES FROM SOURCE-STORAGEGROUP,
32 //* AND THEIR MATCHING TAPES IN THE BACKINFO FILE,
33 //* PAIR WITH TARGET VOLUMES FROM TO-STORAGEGROUP
34 //*
35 //* RESTORE-FROM-DUMPTAPES -
36 12 //* MAX-TAPEDRIVES ( 4 ) -
37 8 //* BACKINFO-DDN( BACKINFO ) -
38 9 //* VOLPAIRS-DDN( VOLPAIRS ) -
39 11 //* USERCATALOGS-DDN( UCATS ) -
40 13 //* SOURCE-STORAGEGROUP (
41 //* SGRPX1 SGRPX2 -
42 //* ) -
43 14 //* TO-STORAGEGROUP (
44 //* SGRPY1 SGRPY2 -
45 //* ) -
46 15 //* USERCATALOGS (
47 //* USERCAT.SRC01 USERCAT.TGT01 -
48 //* USERCAT.SRC02 USERCAT.TGT02 -
49 //* ) -
50 //*
51 //* SAMPLE 2: RESTORE FROM DUMPTAPES USING UP TO 6 TAPE DRIVES,
52 //* USING ALL SOURCE VOLUMES AND MATCHING TAPES
53 //* IN THE BACKINFO FILE, PAIR WITH TARGET VOLUMES
54 //* FROM TO-VOLSER
55 //*
56 //* RESTORE-FROM-DUMPTAPES -
57 12 //* MAX-TAPEDRIVES ( 6 ) -
58 8 //* BACKINFO-DDN( BACKINFO ) -
59 9 //* VOLPAIRS-DDN( VOLPAIRS ) -
60 11 //* USERCATALOGS-DDN( UCATS ) -
61 16 //* TO-VOLSER (
62 //* TGT001 TGT002 TGT003 -
63 //* ) -
64 15 //* USERCATALOGS (
65 //* USERCAT.SRC01 USERCAT.TGT01 -

```

```

//*      USERCAT.SRC02 USERCAT.TGT02  -
//*      )
//*
//

```

1. Deletion of volume pairs data set in anticipation of allocating new for each execution.
2. Deletion of the status data set in anticipation of allocating new for each execution.
3. Deletion of user catalog data set in anticipation of allocating new for each execution.
4. Execution of Db2 Cloning Tool main program.
5. Db2 Cloning Tool LOAD library must be authorized.
6. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
7. DD for CKZPRINT output.
8. The backinfo data set that is created by the Db2 Cloning Tool DB2GETBACKINFO command from the source Db2 system. It contains information about the Db2 system level backup information from HSM LIST COPYPOOL, and will be used as input to the RESTORE-FROM-DUMPTAPES command.
9. Output DD for volume pairing information for use in later COPY(PGM(NONE)) step.
10. DD to record the status and progress of RESTORE-FROM-DUMPTAPES command. This DD is required for RERUN.
11. Output DD for user catalog information for use in later COPY(PGM(NONE)) step.
12. Specifies the maximum available number of tape drives available for tape allocation (and implicitly, the number of restore subtasks).
13. Specifies the source SMS storage groups.
14. Specifies the target SMS storage groups that will be used to pair the target volumes with the dump volumes. Storage groups are matched one-for-one with the SOURCE-STORAGEGROUP storage groups.
15. Lists the source catalogs that contain data sets from source volumes, and the corresponding target catalog for renamed volume data sets.
16. Specifies the target volumes that are to be paired with dump volumes.

UCATOPTIONS

This command is not required. UCATOPTIONS BACKUP will back up source catalogs when the COPY command used the USERCATALOGS-NOBACKUP keyword. If VOLSER was specified for the source catalog when executing COPY with the USERCATALOGS-NOBACKUP keyword, the backup is taken from the copies of the source catalogs that reside on target volumes. UCATOPTIONS BACKUP must be run before the RENAME command.

UCATOPTIONS LIST will list the source and target ICF catalog pairs and CATWORK data set names in the Db2 Cloning Tool journal.

UCATOPTIONS UPDATE will allow the target user catalog name(s) or CATWORK data set name(s) to be changed.

For example, if you specify a target catalog(s) in the COPY command, and want to change the target catalog(s) used during the RENAME command, this command will list the source and target ICF catalog pairs currently in the journal and allow you to update the journal with the new target ICF catalog names.



Attention: If the COPY command used the USERCATALOGS-NOBACKUP keyword, the catalog backup can be done by using the UCATOPTIONS command with the BACKUP keyword. However, the source ICF catalog information being backed up needs to be in synchronization with the contents of the copied volumes. If UCATOPTIONS BACKUP is used (meaning that USERCATALOGS-NOBACKUP was specified for the COPY command), the user must ensure that the source ICF catalogs are logically at the same point in time as when the volumes were copied.

UCATOPTIONS command syntax

UCATOPTIONS

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
BACKUP [ FORCE ] | LIST | UPDATE
```

If UPDATE is specified, one of the following keywords must be specified:

```
NEWCATWORKS( current-dsn1, new-dsn1, current-dsn2, new-dsn2, ... ) |  
NEWCATWORKS-DDN( ddname )  
NEWTARGETS( srcusercatalog1, newtgtusercatalog1,  
             srcusercatalog2, newtgtusercatalog2, ... ) |  
NEWTARGETS-DDN( ddname )
```

UCATOPTIONS command and keyword definitions

Required keywords are described first, followed by optional keywords.

UCATOPTIONS

Optional command that will either list the user catalog pairs from the Db2 Cloning Tool journal, or allow the target user catalog name(s) to be changed, or back up source catalogs.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a ddname assumed via the JCL to point to a journal data set. The journal is used to pass information between Db2 Cloning Tool steps.

- Default: None
- Required: Yes
- Restrictions: None

BACKUP [FORCE] | LIST | UPDATE

BACKUP requests that the source catalogs be backed up. The backup was not done by COPY because the USERCATALOGS-NOBACKUP keyword was used. If VOLSER was specified for the source catalog when executing COPY with the USERCATALOGS-NOBACKUP keyword, the backup is taken from copies of the source catalogs that reside on target volumes.

UCATOPTIONS BACKUP must be run before the RENAME command. If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN were used with USERCATALOGS-NOBACKUP in the COPY, VOLOPTIONS TARGETOFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

BACKUP FORCE requests that the source catalogs be backed up even if they have already been backed up.

LIST requests a display of the current source and target ICF catalog pairs.

UPDATE changes the existing target ICF catalog entries found in the journal with the new target ICF catalog names. UPDATE can not be used if the catalog backup has not been done.

Note: The existing source ICF catalogs must be entered in pairs with the new target ICF catalogs as shown in the NEWTARGETS and NEWTARGETSDDN keyword description that follows.

- Default: LIST
- Required: No
- Restrictions: None for LIST or BACKUP. UPDATE requires one of the following: NEWCATWORKS, NEWCATWORKS-DDN, NEWTARGETS, or NEWTARGETSDDN.

**NEWCATWORKS (*current-dsn1*, *new-dsn1*, *current-dsn2*, *new-dsn2*, ...)
or **NEWCATWORKS-DDN(*ddname*)****

NEWCATWORKS specifies the current CATWORK data set name and the new catwork data set name. The current CATWORK data set name must match an Db2 Cloning Tool journal entry.

new-dsn1 will replace the current CATWORK data set name with the value of *current-dsn1*, with the new CATWORK data set name in the journal entry.

NEWCATWORKS-DDN specifies a DD name assumed via the JCL to point to a data set containing the current and new CATWORK data set name pairs. The pairs are the same format as in the NEWCATWORKS keyword.

- Default: None
- Required: No
- Restrictions: May only be specified with UPDATE.

**NEWTARGETS(*srcusercatalog1*, *newtgtusercatalog1*, *srcusercatalog2*, *newtgtusercatalog2*, ...)
or **NEWTARGETS-DDN(*ddname*)****

NEWTARGETS specifies the current source ICF catalog and the new target ICF catalog. The current source ICF catalog must match an Db2 Cloning Tool journal entry.

newtgtusercatalog1 will replace the current target user catalog which is paired with *srcusercatalog1* in the journal entry.

NEWTARGETS-DDN specifies a DD name assumed via the JCL to point to a data set containing the source and target ICF catalog pairs. The pairs are the same format as in the NEWTARGETS keyword.

- Default: None
- Required: No
- Restrictions: May only be specified with UPDATE.

UCATOPTIONS step JCL example

This topic contains an example of UCATOPTIONS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZUCATO.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The UCATOPTIONS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1  //??????? JOB , 'UCATOPTIONS' , CLASS=A,MSGCLASS=X
2  //S1      EXEC PGM=CKZ00010,REGION=8M
3  //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
4  //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI) ,DISP=SHR
5  //CKZPRINT DD SYSOUT=*
   //SYSUDUMP DD SYSOUT=*
6  //JOURNAL DD DSN=CKZ.JRNL,DISP=OLD
   //CKZIN  DD *
   UCATOPTIONS
7  JOURNAL-DDN(JOURNAL)
   //*

```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.

4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the UCATOPTIONS command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

VARYOFF

This command is not required. VARYOFF is provided to vary target or source volumes offline on only the local system where the command is run, on all systems in the sysplex, or on all systems in the sysplex other than the local system.

Note: VARYOFF should not be used in JES3 environment. The VARYOFF command issues an MVS VARY command, which is not recommended for JES3-managed devices.

The volumes to be varied offline can be specified by:

- Using the vary file. The volumes are determined when the VARYOFF command is run by reading a provided COPY command (COPY-CMD-DDN(COPYCMD) DD that points to a file which contains Db2 Cloning Tool COPY command syntax. The COPY syntax is read for volume information but is not executed during the VARYOFF processing). The volumes are then saved in the vary file for use by a subsequent VARYON command to bring the volumes online.

The vary file can be used when the volumes have different device numbers on different systems. For this case the VARYOFF command would be run with SCOPE(LOCAL) on each system with a separate vary file for each system. See the example using the vary off file in the topic [“VARYOFF step JCL example” on page 501](#).

- Using an existing journal data set. The volumes have been predetermined by the Db2 Cloning Tool COPY command that created the journal and can be used if the volumes will be varied offline later in the process.

When using SCOPE(SYSPLEX(ALL)), SCOPE(SYSPLEX(OTHER)), or SCOPE(SYSPLEX(SYSGRPNAME(*sysgrpname*))), the volumes need to have the same device numbers on the systems in the sysplex.

VARYOFF command syntax

VARYOFF

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) |
  VARY-DDN( ddname ) }
TARGET | SOURCE
```

Required only if VARY-DDN(*ddname*) is specified:

```
COPY-CMD-DDN( ddname )
```

Optional keywords:

```
MAX-VOLS-PER-CMD( nn | 8 )
SCOPE( LOCAL | SYSPLEX( { ALL | OTHER | SYSGRPNAME(sysgrpname) } [ , T( nnn ) ] ) )
SIMULATE
VOL-ALREADY-OFFLINE( { QUIT | CONTINUE } [ , RC( nnn | 8 ) ] )
```

VARYOFF command and keyword definitions

Required keywords are described first, followed by optional keywords.

VARYOFF

Optional command to vary volumes offline.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point to a journal data set. The journal is used to pass information between Db2 Cloning Tool steps.

- Default: None
- Required: Yes, if VARY-DDN is not specified.
- Restrictions: Mutually exclusive with VARY-DDN.
- Short form(s): JDSN, JDDN

TARGET | SOURCE

Specifies the volume set to use. **TARGET** specifies that the target volumes will be used. **SOURCE** specifies that the source volumes will be used.

- Default: None
- Required: Yes
- Restrictions: None

VARY-DDN(*ddname*)

This parameter supplies the DD name of the Db2 Cloning Tool vary file assumed via the JCL to point at a vary data set.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same vary data set. Each Db2 Cloning Tool volume group needs a different vary data set.

The vary file is used to pass VOLSERS and device numbers between an Db2 Cloning Tool VARYOFF step and an Db2 Cloning Tool VARYON step. Therefore, as noted in the JCL comments, it must be cataloged in the VARYOFF step when first created, and referenced in the VARYON step.

- Default: None
- Required: Yes, if JOURNAL-DSN and JOURNAL-DDN are not specified.
- Restrictions: Mutually exclusive with JOURNAL-DSN and JOURNAL-DDN.

COPY-CMD-DDN(*ddname*)

This parameter supplies the DD name of the Db2 Cloning Tool copy command.

- Default: None
- Required: Yes, if VARY-DDN is specified.
- Restrictions: None

MAX-VOLS-PER-CMD(*nn* | **8**)

Specifies the maximum number of volumes that will be used in a single SYSPLEX vary offline command.

- Default: 8
- Required: no
- Restrictions: Used only when SCOPE(SYSPLEX(..)) is specified.

SCOPE(LOCAL | SYSPLEX({ ALL | OTHER | SYSGRPNAME(*sysgrpname*) } [, T(*nnn*)]))

Specifies the scope of the vary offline commands.

LOCAL specifies that the vary offline commands will be issued only on the local system.

SYSPLEX specifies that the vary offline commands will be issued sysplex-wide.

ALL specifies that the volumes will be varied offline to the local system and vary offline commands will be routed to the systems according to the value of the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If SYSPLEX_GROUPNAME_ALL is set to *ALL or not defined, then the volumes will be varied offline to all other systems in the sysplex. If SYSPLEX_GROUPNAME_ALL is a system group name, then the volumes will be varied offline to all systems of the system group.

OTHER specifies that vary offline commands will be routed to the systems according to the value of the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If SYSPLEX_GROUPNAME_ALL is set to *ALL or not defined, then the volumes will be varied offline to all other systems in the sysplex. If SYSPLEX_GROUPNAME_ALL is a system group name, then the volumes will be varied offline to all systems of this system group, excluding the local system. No vary offline will be issued on the local system.

SYSGRPNAME(sysgrpname) specifies that vary offline commands will be routed to the systems that are defined in the system group *sysgrpname*. If the local system is defined in the system group, then it is also varied offline. The SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member is ignored.

T specifies an optional timeout interval that will be added to the sysplex vary commands.

- Default: Local
- Required: No
- Restrictions: None

SIMULATE

Simulate will verify the syntax and determine the volumes to be varied offline but will not issue any vary offline commands.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

VOL-ALREADY-OFFLINE({ QUIT | CONTINUE } [, (RC(*nnn* | 8))])

Specifies the action to be taken when a volume to be processed is already offline.

QUIT specifies that processing will quit when the first volume already offline is encountered.

CONTINUE specifies that processing will continue with the next volume when a volume already offline is encountered.

RC specifies the return code that will be used when a volume already offline is encountered.

- Default: QUIT, RC(8)
- Required: No
- Restrictions: Used only when SCOPE(LOCAL) or SCOPE(SYSPLEX(ALL)) are specified, or when SCOPE(SYSPLEX(SYSGRPNAME(...))) is specified and the local system is defined in the system group.

VARYOFF step JCL example

This topic contains several examples of VARYOFF step JCL.

VARYOFF step JCL example - varying volumes offline using vary file

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVOFFV.

The VARYOFF step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//??????? JOB , 'VARYOFF', CLASS=A, MSGCLASS=X
//S0 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
1 DEL CKZ.VARY
  SET MAXCC=0
2 //S1 EXEC PGM=CKZ00010, REGION=6M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
6 //VARY DD DSN=CKZ.VARY, RECOG=KS, KEYLEN=64, KEYOFF=0,
// DISP=(, CATLG), UNIT=SYSALLDA,
// LRECL=600, SPACE=(CYL, (10, 10))
//CKZIN DD *
  VARYOFF -
7 TARGET -
8 SCOPE (LOCAL) -
9 COPY-CMD-DDN (COPYCMD) -
6 VARY-DDN (VARY)
//*
9 //COPYCMD DD *
  COPY -
  DATA-MOVER( -
    COPYCMDLIMIT (24) -
  ) -
  FROM-VOLSER (VSRC02) -
7 TO-VOLSER (VTGT02) -
  USERCATALOGS( -
    USERCAT .SRC01 USERCAT .TGT01 -
    USERCAT .SRC02 USERCAT .TGT02 -
  ) -
  CATWORK-DSN (CKZ.WRK.* ) -
  JOURNAL-DDN (JOURNAL)
//*

```

1. Deletion of vary data set in anticipation of allocating new for each execution.
2. Execution of Db2 Cloning Tool main program.
3. Db2 Cloning Tool LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Vary data set. This VSAM data set is used to pass information between the Db2 Cloning Tool VARYOFF and VARYON steps. The vary file is allocated and cataloged in the VARYOFF step and used as input to the VARYON command. In the sample JCL, the control statement VARY-DDN(VARY) is specifying that a DD with the name VARY is being used.
7. TARGET parameter that specifies the target volumes found in the journal will be varied offline. In the COPY command, the target volumes specified will be used.
8. SCOPE parameter specifying LOCAL will cause the vary offline commands to be issued only on the local system.
9. The COPY command. The COPY command is used to derive the volumes for processing. In the sample JCL, the control statement COPY-CMD-DDN(COPYCMD) is specifying that a DD with the name COPYCMD is being used to read the copy command.

VARYOFF step JCL example - varying volumes offline using journal

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVOFFJ.

The VARYOFF step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//??????? JOB , 'VARYOFF', CLASS=A, MSGCLASS=X
1 //S1 EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
//CKZIN DD *
    VARYOFF -
6     TARGET -
7     SCOPE(SYSPLEX(OTHER)) -
5     JOURNAL-DDN(JOURNAL)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the VARYOFF command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) is specifying that a DD with the name JOURNAL is being used rather than a data set name.
6. TARGET parameter that specifies the target volumes found in the journal will be varied offline
7. SCOPE parameter specifying SYSPLEX(OTHER) will cause the vary offline commands to be routed to all other systems in the sysplex.

VARYON

This command is not required. VARYON is provided to vary target or source volumes online. The VARYON command can vary volumes online on only the local system where the command is run, on all systems in the sysplex, or on all systems in the sysplex other than the local system.

When using SCOPE(SYSPLEX(ALL)) or SCOPE(SYSPLEX(OTHER)) the volumes need to have the same device numbers on the systems in the sysplex.

Note: VARYON should not be used in JES3 environment. The VARYON command issues an MVS VARY command, which is not recommended for JES3-managed devices.

The volumes to be varied online can be specified by:

- Using the vary file. By using the vary file, the volumes have been predetermined by the Db2 Cloning Tool VARYOFF command that created the vary file. The vary file can be used when the volumes have different device numbers on different systems. For this case, the VARYON command would be run with SCOPE(LOCAL) on each system with a separate vary file for each system.
- Using an existing journal data set. The volumes have been predetermined by the CKZ COPY command that created the journal.

When using SCOPE(SYSPLEX(ALL)), SCOPE(SYSPLEX(OTHER)), or SCOPE(SYSPLEX(SYSGRPNAM(*sysgrpname*))), the volumes need to have the same device numbers on the systems in the sysplex.

VARYON command syntax

VARYON

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) |  
  VARY-DDN( ddname ) }
```

Required only if **JOURNAL-DSN(data set name)** or **JOURNAL-DDN(ddname)** is specified:

```
TARGET | SOURCE
```

Optional keywords:

```
MAX-VOLS-PER-CMD( nn | 8 )  
SCOPE( LOCAL | SYSPLEX( { ALL | OTHER | SYSGRPNAME( sysgrpname ) } [ , T( nnn ) ] ) )  
SIMULATE  
VOL-ALREADY-ONLINE( { QUIT | CONTINUE } [ , ( RC( nnn | 8 ) ) ] )  
WRONG-VOLSER( { QUIT | CONTINUE } [ , ( RC( nnn | 8 ) ) ] )
```

VARYON command and keyword definitions

Required keywords are described first, followed by optional keywords.

VARYON

Optional command to vary volumes online.

- Required: No
- Restrictions: None

JOURNAL-DSN(data set name) or **JOURNAL-DDN(ddname)**

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point to a journal data set. The journal is used to pass information between Db2 Cloning Tool steps.

- Default: None
- Required: Yes, if VARY-DDN is not specified.
- Restrictions: Mutually exclusive with VARY-DDN.
- Short form(s): JDSN, JDDN

TARGET | SOURCE

Specifies the volume set to use. **TARGET** specifies that the target volumes will be used. **SOURCE** specifies that the source volumes will be used.

- Default: None
- Required: Yes, if JOURNAL-DSN or JOURNAL-DDN is specified.
- Restrictions: Mutually exclusive with VARY-DDN.

VARY-DDN(ddname)

This parameter supplies the DD name of the Db2 Cloning Tool vary file assumed via the JCL to point at a vary data set.

If multiple Db2 Cloning Tool setups are used for different volume groups, DO NOT use the same vary data set. Each Db2 Cloning Tool volume group needs a different vary data set.

The vary file is used to pass VOLSERs and device numbers between an Db2 Cloning Tool VARYOFF step and an Db2 Cloning Tool VARYON step. Therefore, as noted in the JCL comments, it must be cataloged in the VARYOFF step when first created, and referenced in the VARYON step.

- Default: None
- Required: Yes, if JOURNAL-DSN and JOURNAL-DDN are not specified.
- Restrictions: Mutually exclusive with JOURNAL-DSN and JOURNAL-DDN.

MAX-VOLS-PER-CMD(*nn* | 8)

Specifies the maximum number of volumes that will be used in a single SYSPLEX vary online command.

- Default: 8
- Required: no
- Restrictions: Used only when SCOPE(SYSPLEX(..)) is specified.

SCOPE(LOCAL | SYSPLEX({ ALL | OTHER | SYSGRPNAME(*sysgrpname*) } [, T(*nnn*)]))

Specifies the scope of the vary online commands.

LOCAL specifies that the vary online commands will be issued only on the local system.

SYSPLEX specifies that the vary online commands will be issued sysplex-wide.

ALL specifies that the volumes will be varied online to the local system and vary online commands will be routed to the systems according to the value of the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If SYSPLEX_GROUPNAME_ALL is set to *ALL or not defined, then the volumes will be varied online to all other systems in the sysplex. If SYSPLEX_GROUPNAME_ALL is a system group name, then the volumes will be varied online to all systems of the system group.

OTHER specifies that vary online commands will be routed to the systems according to the value of the SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member. If SYSPLEX_GROUPNAME_ALL is set to *ALL or not defined, then the volumes will be varied online to all other systems in the sysplex. If SYSPLEX_GROUPNAME_ALL is a system group name, then the volumes will be varied online to all systems of this system group, excluding the local system. No vary online will be issued on the local system.

SYSGRPNAME(*sysgrpname*) specifies that vary online commands will be routed to the systems that are defined in the system group *sysgrpname*. If the local system is defined in the system group, then it is also varied online. The SYSPLEX_GROUPNAME_ALL token in the CKZINI PARMLIB member is ignored.

T specifies an optional timeout interval that will be added to the sysplex vary commands.

- Default: Local
- Required: No
- Restrictions: None

SIMULATE

Simulate will verify the syntax and determine the volumes to be varied offline but will not issue any vary online commands.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

VOL-ALREADY-ONLINE({ QUIT | CONTINUE } [, (RC(*nnn* | 8))])

Specifies the action to be taken when a volume to be processed is already online.

QUIT specifies that processing will quit when the first volume already offline is encountered.

CONTINUE specifies that processing will continue with the next volume when a volume already online is encountered.

RC specifies the return code that will be used when a volume already online is encountered.

- Default: QUIT, RC(8)
- Required: No
- Restrictions: Used only when SCOPE(LOCAL) or SCOPE(SYSPLEX(ALL)) are specified, or when SCOPE(SYSPLEX(SYSGRPNAME(...))) is specified and the local system is defined in the system group.

WRONG-VOLSER({ QUIT | CONTINUE } [, (RC(nnn | 8)])

Specifies the action to be taken when a volume that is brought online has a different VOLSER than expected.

QUIT specifies that processing will quit when the first volume with a wrong VOLSER is encountered.

CONTINUE specifies that processing will continue with the next volume when a volume with a wrong VOLSER is encountered.

RC specifies the return code that will be used when a volume with a wrong VOLSER is encountered.

- Default: QUIT, RC(8)
- Required: No
- Restrictions: Used only when SCOPE(LOCAL) or SCOPE(SYSPLEX(ALL)) are specified.

VARYON step JCL example

This topic contains several examples of VARYON step JCL.

VARYON Step JCL – Example to vary volumes online using vary file

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVONV.

The VARYON step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'VARYON' , CLASS=A , MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010 , REGION=6M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD , DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) , DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
6 //VARY DD DSN=CKZ.VARY , DISP=OLD
5 //CKZIN DD *
 VARYON
6 SCOPE(LOCAL)
 VARY-DDN(VARY)
//*
```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Vary data set. This VSAM data set is used to pass information between the Db2 Cloning Tool VARYOFF and VARYON steps. The vary file is allocated and cataloged in the VARYOFF step and used as input to the VARYON command. In the sample JCL, the control statement VARY-DDN(VARY) is specifying that a DD with the name VARY is being used.
6. SCOPE parameter specifying LOCAL will cause the vary online commands to be issued only on the local system.

VARYON Step JCL – Example to vary volumes online using journal

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVONJ.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The VARYON step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//??????? JOB , 'VARYON' , CLASS=A , MSGCLASS=X
1 //S1 EXEC PGM=CKZ00010 , REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD , DISP=SHR
3 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI) , DISP=SHR
4 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL , DISP=OLD
//CKZIN DD *
    VARYON
6     TARGET -
7     SCOPE(SYSPLEX(OTHER)) -
5     JOURNAL-DDN(JOURNAL)
//*

```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the VARYON command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) is specifying that a DD with the name JOURNAL is being used rather than a data set name.
6. TARGET parameter that specifies the target volumes found in the journal will be varied online.
7. SCOPE parameter specifying SYSPLEX(OTHER) will cause the vary online commands to be routed to all other systems in the sysplex.

VOLOPTONS

This command is not required. VOLOPTONS is intended for situations where the Db2 Cloning Tool COPY step is run at one site (SITEA) and the Db2 Cloning Tool RENAME step is run at another site (SITEB). VOLOPTONS LIST, SOURCECLIP, and UPDATE are intended to support such scenarios.

VOLOPTONS can also be used when the target volumes are offline to the Db2 Cloning Tool COPY step and it is not desirable for Db2 Cloning Tool COPY to re-label and vary online the target volumes. For instance, the target offline volumes will be backed up prior to the Db2 Cloning Tool RENAME step being run. For this function, COPY would use the VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN keywords, which cause COPY to not re-label or vary online the target volumes. The offline target volumes are then backed up, and VOLOPTONS TARGETOFFLINECLIP is used to re-label and vary online the target volumes prior to running the RENAME command.

VOLOPTONS can also be used when the target volumes are online and it is desired to vary them offline and relabel them to their corresponding source volume volser. VOLOPTONS TARGETUNCLIP can be used to support this scenario.

For example: SITEA runs the Db2 Cloning Tool COPY which FlashCops (or snaps) source volumes SRC001, SRC002, and SRC003 to target volumes TGT001, TGT002, and TGT003. SITEA then dumps TGT001, TGT002, TGT003 to tape and sends them to SITEB. The COPY also backs up the source user catalogs and establishes the source/target user catalog relationship.

In addition, SITEA backs up the Db2 Cloning Tool journal file and the data sets created under the CATWORK-DSN prefix. The journal file and CATWORK-DSN data sets are created by the Db2 Cloning Tool COPY command.

SITEB restores the Db2 Cloning Tool journal file and the CATWORK-DSN data sets on their system. The CATWORK-DSN data set names must match those used by SITEA. The journal file and CATWORK-DSN data sets must be cataloged.

SITEB restores the tape(s) using 'ADRDSSU RESTORE COPYVOLID'.

At this point, SITEB has SRC001, SRC002, and SRC003 online to their system. The data sets on these volumes are not cataloged.

In order for Db2 Cloning Tool RENAME to rename and catalog the SITEB data sets, the volumes need to be clipped to the corresponding target volume serials.

VOLOPTIONS SOURCECLIP will do this function. It will vary SRC001, SRC002, and SRC003 offline on the image where the command is executed. It will then use ICKDSF to change SRC001 to TGT001, SRC002 to TGT002, and SRC003 to TGT003. The target volumes will be varied online by VOLOPTIONS SOURCECLIP to the image where the command is executed.

If SITEB needs to see the volume pairs used by SITEA, VOLOPTIONS LIST will display the current source/target volume pairs.

Db2 Cloning Tool uses the Db2 Cloning Tool journal to identify and communicate source and target volume pairs to Db2 Cloning Tool commands. If SITEB needs to use different target volume serials than those used at SITEA, VOLOPTIONS UPDATE NEWTARGETS will allow the current target volume serials in the Db2 Cloning Tool journal to be changed.

VOLOPTIONS LIST could be used to provide the initial input for the NEWTARGETS keyword. The new target volume serial(s) would need to be added to the source/target pairs from the VOLOPTIONS LIST command.

VOLOPTIONS UPDATE NEWTARGETS and NEWTARGETSDEVN do not support changing any target volume serial to a source volume serial.

VOLOPTIONS command syntax

VOLOPTIONS

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
LIST | SOURCECLIP | TARGETOFFLINECLIP | TARGETUNCLIP | UPDATE  
RESUME  
SIMULATE
```

Required only if UPDATE is specified:

```
{ NEWTARGETS( srcvolser1 tgtvolser1 newtgtvolser1 ... [ , srcvolsern  
tgtvolsern newtgtvolsern ] ) |  
NEWTARGETS-DDN( ddname ) |  
NEWTARGETSDEVN  
( srcvolser1 tgtvolser1 newtgtvolser1 newtgtdevn1 ... [ , srcvolsern tgtvolsern  
newtgtvolsern newtgtdevnn ] ) |  
NEWTARGETSDEVN-DDN( ddname ) }
```

VOLOPTIONS command and keyword definitions

Required keywords are described first, followed by optional keywords.

VOLOPTIONS

Optional command to use when the COPY command is run at one site and the RENAME command is run at another.

- Required: No
- Restrictions: None

**JOURNAL-DSN (data set name)
or JOURNAL-DDN (ddname)**

This parameter supplies either the data set name of the Db2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple CKZ setups are used for different volume groups, DO NOT use the same journal data set. Each CKZ 'application' needs a different journal data set.

The journal is used to pass information between CKZ steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None
- Short form(s): JDSN, JDDN

LIST | SOURCECLIP | TARGETOFFLINECLIP | TARGETUNCLIP | UPDATE

LIST requests a display of the current source volume serial/target volume serial pairs.

SOURCECLIP requests that the online source volume serials be changed with ICKDSF to their paired target volume serials.

TARGETOFFLINECLIP requests that the offline target volumes which currently have source volume serials have their volume serials changed with ICKDSF to their target volume serials and varied online. If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN was used with USERCATALOGS-NOBACKUP in the COPY, VOLOPTIONS TARGETOFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

TARGETUNCLIP requests that the online target volumes be varied offline and the target serials be changed with ICKDSF to their paired source volume serials.

UPDATE requests changing specified target volume serials to new target volume serials.

- Default: LIST
- Required: No
- Restrictions: UPDATE requires NEWTARGETS or NEWTARGETSDEVN. No restrictions for LIST, SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP.

RESUME

RESUME specifies that SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP should resume processing of any volumes that failed to be completely processed by a previous SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP command.

Prior to running SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP with RESUME, the problem that caused a volume to not be completely processed by SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP should be resolved.

- Default: None
- Required: No
- Restrictions: This keyword can only be specified if SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP is specified.

SIMULATE

Simulate will verify the syntax and determine the volumes to be processed and will display what action would have been taken but will not vary any volumes offline or online or change any volume serials with ICKDSF or update the journal.

- Default: None

- Required: No
- Restrictions: None
- Short form: SIM

NEWTARGETS (*srcvolser1 tgtvolser1 newtgtvolser1 ... [, srcvolsern tgtvolsern newtgtvolsern]*)
 or **NEWTARGETS-DDN** (*ddname*) | **NEWTARGETSDEVN** (*srcvolser1 tgtvolser1 newtgtvolser1 newtgtdevn1 ... [srcvolsern tgtvolsern newtgtvolsern newtgtdevnn]*)
 or **NEWTARGETSDEVN-DDN** (*ddname*)

NEWTARGETS specifies the current source volume serial, the current target volume serial, and the new target volume serial. The current source volume serial and current target volume serial must match the CKZ journal entries. For example: 'newtgtvolser1' will replace 'tgtvolser1' in the journal entry. The short form is NTGT.

NEWTARGETS-DDN specifies a DD name which has a file containing the volume serial triplets. The triplets are the same format as in the NEWTARGETS keyword.

NEWTARGETSDEVN specifies the current source volume serial, the current target volume serial, and the new target volume serial and new target device number. The current source volume serial and current target volume serial must match the CKZ journal entries. For example: 'newtgtvolser1' will replace 'tgtvolser1' and 'newtgtdevn1' will replace the target device number in the journal entry. The short form is NTGTD.

NEWTARGETSDEVN-DDN specifies a DD name which has a file containing the volume serial and devn quadruplets. The quadruplets are the same format as in the NEWTARGETSDEVN keyword.

- Default: None
- Required: One of the NEWTARGETS keywords is required if UPDATE is specified.
- Restrictions: May only be specified with UPDATE.

VOLOPTIONS step JCL example

This topic contains an example of VOLOPTIONS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZVOLOP.

For completeness and to illustrate where JCL parameters and Db2 Cloning Tool control statements must match, the following JCL includes sample Db2 Cloning Tool control statements.

The VOLOPTIONS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'VOLOPTIONS', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
7 //CKZIN DD *
8 VOLOPTIONS
9 JOURNAL- DD JOURNAL
10 //*

```

1. Execution of Db2 Cloning Tool main program.
2. Db2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the Db2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by Db2 Cloning Tool commands. This VSAM data set is used to pass information between Db2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the VOLOPTIONS command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

The previous JCL example used LIST, the default, so no keyword was specified. The following two examples are provided for SOURCECLIP and UPDATE keywords:

Example using VOLOPTIONS (SOURCECLIP):

```
VOLOPTIONS SOURCECLIP -  
JOURNAL-DDN(JOURNAL)
```

Example using VOLOPTIONS (UPDATE):

```
VOLOPTIONS UPDATE  
  NEWTARGETS(VSRC01,VTGT11,VTGT01 -  
             VSRC02,VTGT22,VTGT02 -  
             VSRC03,VTGT13,VTGT03) -  
JOURNAL-DDN(JOURNAL)
```


Chapter 27. Db2 Cloning Tool Table Space Cloning commands

All Db2 Cloning Tool Table Space Cloning commands are invoked by executing the main program, CKZ00500. Functionality is selected by specifying the appropriate Db2 Cloning Tool Table Space Cloning command and parameters.

Most Db2 Cloning Tool Table Space Cloning commands support and modify the behavior of the COPY command.

The Db2 Cloning Tool Table Space Cloning commands are listed in the following table and also explained in detail in separate topics.

| Db2 Cloning Tool Table Space Cloning Command | Description |
|--|--|
| “COPY” on page 513 | Controls all phases of replicating the Db2 table spaces and index spaces that are to be cloned. |
| “DATASUBTYPE” on page 560 | Optional command that may be used to specify the data subtype value (BIT, MIXED, or SBCS) needed by data masking. |
| “HLQDDDF” on page 561 | Optional command allows Db2 Cloning Tool Table Space Cloning to pass input and output DDs to ADRDSSU or the EMC API. |
| “LISTDEF” on page 562 | Used to select to source Db2 table spaces and index spaces to be copied to the target Db2. |
| “SET” on page 567 | Specifies the local Db2 subsystem for the source job and the TCP/IP server job. |

COPY

The COPY command is required. The method that Db2 Cloning Tool Table Space Cloning uses to copy the data sets from source to target is controlled by the DATA-MOVER command. In all cases, COPY captures the Db2 catalog data pertaining to source and target data sets.

Depending upon how the COPY subcommand DATA-MOVER is specified, one of the following processes occurs:

- If DATA-MOVER(PGM(ADRDSSU)) is specified, COPY invokes FlashCopy or SnapShot to perform data set copies.
- If DATA-MOVER(PGM(EMCAPI)) is specified, COPY invokes data set copies via TimeFinder/Clone Mainframe Snap Facility's data set level support facility.
- If DATA-MOVER(PGM(SRCIMCPY)) is specified, image copies are used as input to the table space cloning process.
- If DATA-MOVER(PGM(SRCVSCPY)) is specified, VSAM data sets are used as the source for the table space cloning process.

Note: SRCVSCPY has been deprecated. Support is likely to be removed in the future. For more information, see [“Deprecated and removed functions and parameters” on page 10](#).

- If DATA-MOVER(PGM(NONE)) is specified, COPY assumes that you will create the data set copies outside of Db2 Cloning Tool Table Space Cloning.

The COPY command process does the following:

1. Connects to the source Db2 catalog specified by the LOCAL-SSID keyword of the SET command

2. Reads the source Db2 catalog to get a list of table spaces and index spaces specified by the LISTDEF DD CKZLSTDF
3. Acquires the attributes and IDs of the source Db2 table spaces and index spaces remaining after LISTDEF processing is complete
4. Builds VSAM data set names from the entries in the source Db2 catalog
5. Queries the target Db2 catalog, specified by the COPY command sub-keyword SSID of keyword TARGET-DB2, and verifies the existence of matching target Db2 table spaces and index spaces
6. Acquires from the target Db2 catalog the attributes and IDs of the existing target Db2 table spaces and index spaces
7. Issues Db2 STOP commands for the source and the target (if requested) if PGM=ADRDUSSU, EMCAPI, SRCIMCPY, or SRCVSCP
8. Invokes FlashCopy or SnapShot (if available) if PGM=ADRDUSSU; TimeFinder/Clone if PGM=EMCAPI; processes source image copies if PGM=SRCIMCPY; or processes VSAM data sets as the source if PGM=SRCVSCP
9. Issues Db2 START commands for the source if PGM=ADRDUSSU, EMCAPI, SRCIMCPY, or SRCVSCP
10. Outputs target commands to the SYNCDB2 data set for input to the target job

COPY command syntax

COPY

Required Keywords:

TARGET-DB2(SSID(*sourcesubsystem*))
 conditionally required keywords . . .
 optional keywords . . .

Conditionally Required Keywords:

TARGET-DB2
 DEFVCAT(*targethlq*)
 IP(*ppp.ppp.ppp.ppp*)
 NAME(*name*)
 PORT(*port*)

Optional Keywords:

```
[ ALLOW-COPY-ON-MISMATCH( Y | N ) ]
[ ALWAYS-COPY-HISTORY-TABLES ( Y | N ) ]
[ ALWAYS-COPY-INDEXTABLES ( Y | N ) ]
[ AUTO-INDEXTABLE-TRANSLATE( Y | N ) ]
[ AUTO-START-SOURCE-SPACE( Y | N | R ) ]
[ AUTO-START-TARGET-SPACE( Y | N ) ]
[ AUTO-STOP-TARGET-SPACE( Y | N ) ]
[ AUTO-TABLESPACE-TRANSLATE( Y | N ) ]
[ CATALOG-PREFETCH(
  [ENABLE-PREFETCH ( Y | N ) ]
  [ENABLE-SOURCE-PREFETCH ( Y | N ) ]
  [ENABLE-TARGET-PREFETCH ( Y | N ) ]
  [READ-FROM-SERVER-CACHE( Y | N ) ]
  [TARGET-PREFETCH-DATABASE-LIST ( targetdatabase1, targetdatabase2,
    . . . targetdatabasesn ) ]
) ]
[ CHECK-DATASET-COMPATIBILITY( Y | N ) ]
[ CHECK-INDEX-KEYS( Y | N ) ]
[ COPY-IF-NO-DB2-TARGET-OBJECTS( Y | N ) ]
[ COPY-IJ-TO-NONEXISTENT-TARGET( Y | N ) ]
[ DATA-MASKING ( Y | N ) ]
[ DATA-MOVER( PGM( ADRDUSSU | EMCAPI | SRCIMCPY | SRCVSCP | NONE )
  [ ,CMDDNAME( ddname ) ]
  [ ,DATACLAS( class ) ]
  [ ,FASTREP( PREF | REQ | NONE ) ]
  [ ,FCTOPRCPRIMARY [ (PRESMIRREQ | PRESMIRPREF | PRESMIRNONE ) ]
  [ ,FCTOXRCPPRIMARY ]
```

```

[ ,MGMTCLAS( class ) ]
[ ,NULLSTORCLAS( Y | N ) ]
[ ,REUSE( Y | N ) ]
[ ,STORCLAS( class ) ]
[ ,USE-LAST-CONSISTENT-FLASHCOPY( Y | N ) ]
[ DATASETS-TO-COPY-DDN( ddname ) ]
[ DSNS-PER-COPY( numberofdatasets ) ]
[ DSS-COPY-COMMANDS( numberofcommands ) ]
[ ENCRYPTION-MISMATCH-RC( 0 | 4 | 8 ) ]
[ EXCLUDE-MISMATCH-PROCESSING( Y | N ) ]
[ FUZZY-COPY( Y | N ) ]
[ IDCAMS-DDN( ddname ) ]
[ IGNORE-RF-MISMATCH-IF-NO-VAR-COLS( Y | N ) ]
[ INCLUDE-ALL-RI( Y | N ) ]
[ JOB-TEMPLATE( inddname1, outddname1,...[ , inddnamen, outddnamen ] ) ]
[ LOG-APPLY( LA-ENABLE( Y | N )
[ ,ALLOW-SKIP-INACTIVE-MEMBERS( Y | N ) ]
[ ,COMMON-CONSISTENT-POINT( Y | N ) ]
[ ,DATA-SHARING-MEMBERS( id | ssid | zparm ) ]
[ ,DELETE-MINILOG-DATASETS( Y | N ) ]
[ ,END-POINT( TO_CURRENT | TO_LOGPOINT X'byte_string' | TO_QUIESCE |
TO_TIMESTAMP timestamp ) ]
[ ,GMT-OFFSET( +hh:mm | -hh:mm ) ]
[ ,IMAGE-COPY-PREFERENCE( copy_types ) ]
[ ,INDEX-LOG-APPLY( Y | N ) ]
[ ,MINILOG-HLQ( miniloghighlevelqualifier ) ]
[ ,MINILOG-LARGE-FILE-TYPE( Y | N ) ]
[ ,MINILOG-UNIT-TYPE( SYSALLDA ) ]
[ ,MINILOG-QUANTITY-IN-TRACKS( Y | N ) ]
[ ,MINILOG-PRIMARY-QUANTITY( nnnn ) ]
[ ,MINILOG-SECONDARY-QUANTITY( nnnn ) ]
[ ,MINILOG-VOLUME-COUNT( nnn ) ]
[ ,MINILOG-DATACLAS( dataclas ) ]
[ ,MINILOG-STORCLAS( storclas ) ]
[ ,MINILOG-MGMTCLAS( mgmtclas ) ]
[ ,NUMBER-OF-BUFFERS( nn ) ]
[ ,NUMBER-OF-CHANNEL-PROGRAMS( nn ) ]
[ ,NUMBER-OF-SORTS( nn ) ]
[ ,QUIESCE-POINT( Y | N ) ]
[ ,SKIP-LOG-APPLY( Y | N ) ]
[ ,SKIP-SYSLGRNX( Y | N ) ]
[ ,SORTFILE-LARGE-FILE-TYPE( Y | N ) ]
[ ,SORTFILE-UNIT-TYPE( SYSALLDA ) ]
[ ,SORTFILE-PRIMARY-QUANTITY( nnnn ) ]
[ ,SORTFILE-SECONDARY-QUANTITY( nnnn ) ]
[ ,SORTFILE-VOLUME-COUNT( nnn ) ]
[ ,SORTFILE-DATACLAS( dataclas ) ]
[ ,SORTFILE-STORCLAS( storclas ) ]
[ ,SORTFILE-MGMTCLAS( mgmtclas ) ]
[ ,SORT-PROGRAM( DFSORT | DB2SORT | SYNCSORT ) ]
[ ,SPACES-PER-MINILOG( numberofspaces ) ]
[ ,SPACES-PER-LOG-APPLY-CALL( numberofspacespercall ) ]
[ ,UNIFIED-WARNING( Y | N ) ]
[ ,USE-LOCAL-TIME( Y | N ) ]
[ ,USE-QUIESCE-POINT-FOR-LOGPOINT( Y | N ) ]
[ ,USE-TCPIP( Y | N ) ]
[ ,WARN-IF-SKIP-QUIESCE( Y | N ) ]
[ ,WARN-IF-TS-DEFINED-LOG-NO( Y | N ) ]
[ ,WORKFILE-LARGE-FILE-TYPE( Y | N ) ]
[ ,WORKFILE-UNIT-TYPE( SYSALLDA ) ]
[ ,WORKFILE-QUANTITY-IN-TRACKS( Y | N ) ]
[ ,WORKFILE-PRIMARY-QUANTITY( nnnn ) ]
[ ,WORKFILE-SECONDARY-QUANTITY( nnnn ) ]
[ ,WORKFILE-VOLUME-COUNT( nnn ) ]
[ ,WORKFILE-DATACLAS( dataclas ) ]
[ ,WORKFILE-STORCLAS( storclas ) ]
[ ,WORKFILE-MGMTCLAS( mgmtclas ) ]
[ ,ZPARAM-MEMBER( zparammembername ) ] ) ]
[ LONGVAR-COMPATIBILITY( Y | N ) ]
[ OBJECT-MISMATCH-RETURN-CODE( mismatchtype1,returncode1,
. . . mismatchtypen,returncoden ) ]
[ OBJECT-TRANSLATE(
DATABASE, sourcedatabase, targetdatabase
[DATABASE, sourcedatabasen, targetdatabasen ]
TABLESPACE, sourcetablespace1, targettablesace1 . . .
[TABLESPACE, sourcetablespace1n, targettablesace1n ]
TABLE, sourcetable1, targettable1 . . .
[TABLE, sourcetable1n, targettable1n ]
INDEXSPACE, sourceindexspace1, targetindexspace1 . . .
[INDEXSPACE, sourceindexspace1n, targetindexspace1n ]
INDEX, sourceindex1, targetindex1 . . .
[INDEX, sourceindex1n, targetindex1n ]

```

```

CREATOR, sourcecreator1, targetcreator1 . . .
  [CREATOR, sourcecreatorn, targetcreatorn ] )
VCAT, sourcevcat1, targetvcat1 . . .
  [VCAT, sourcevcatn, targetvcatn ] ) ]
[ PROCESS-DDL( DDL-ENABLE( Y | N )
  [ ,PROCESS-TYPE( [ Y | N | G | X | A] ) ]
  [ ,PROCESS-DDL-DDN( ddname ) ]
  [ ,COMMIT-FREQUENCY ( numberofstatements ) ]
  [ ,DDL-ATTRIBUTE-CHANGE
    (attributename,srcvalue,trgvalue,applytotype,applytoobject)]
  [ ,EXPLODE-OBJECTS( Y | N ) ] ) ]
  [ ,GENERATE-DDL-DEFAULTS( Y | N ) ]
  [ ,IGNORE-CREATE-OBJECT-EXISTS( Y | N ) ]
  [ ,IGNORE-DROP-OBJECT-DOES-NOT-EXIST( Y | N ) ]
  [ ,RECREATE-OBJECTS( [ A | M | N ] ) ]
  [ ,RECREATE-TABLE-RESTRICTED-ON-DROP( Y | N ) ]
  [ ,SUPPRESS-GRANTS( Y | N ) ]
  [ ,SUPPRESS-RI-CONSTRAINTS( Y | N ) ]
  [ ,USE-DDL-SQLID ( sqlid ) ]
) ]
[ RESET-LOGRBA( Y | N | A ) ]
[ RTS-COPY ( RTS-COPY-ENABLE( Y | N )
  [ ,DELETE-RTS-DATASETS( Y | N ) ] ) ]
  [ ,RTS-DETAILS( Y | N ) ]
  [ ,RTSFILE-DATA-SET-HLQ( dsname ) ]
  [ ,RTSFILE-UNIT-TYPE( SYSALLDA ) ]
  [ ,RTSFILE-QUANTITY-IN-TRACKS( Y | N ) ]
  [ ,RTSFILE-PRIMARY-QUANTITY( nnnn ) ]
  [ ,RTSFILE-SECONDARY-QUANTITY( nnnn ) ]
  [ ,RTSFILE-DATACLAS( dataclas ) ]
  [ ,RTSFILE-STORCLAS( storclas ) ]
  [ ,RTSFILE-MGMTCLAS( mgmtclas ) ]
[ SIMULATE( A | N | Y ) ]
[ START-SOURCE-DDN( ddname ) ]
[ STOP-SOURCE-DDN( ddname ) ]
[ STOP-TARGET-DDN( ddname ) ]
[ SYNCDB2-DDN( ddname ) ]
[ TARGET-JOB-INDEX-REBUILD-DDN( ddname ) ]
[ TARGET-JOB-REPAIR-DDN( ddname ) ]
[ UNLOAD-LOAD ( UNLOAD-LOAD-ENABLE( Y | N )
  [ ,DELETE-DATASETS( Y | N ) ]
  [ ,LOAD-REUSE( Y | N ) ]
  [ ,LOAD-SORTNUM(integer) ]
  [ ,UNLOAD-LOAD-EARLY-VALIDATION ( Y | N ) ]
  [ ,TEMPLATE-SORTOUT-DDN(ddname)
  [ ,TEMPLATE-SYSPUNCH-DDN(ddname)
  [ ,TEMPLATE-SYSREC-DDN(ddname)
  [ ,TEMPLATE-SYSUT-DDN(ddname) ) ]
[ WARN-IF-OBJECT-NOT-TRANSLATED( Y | N ) ]
[ WARN-ON-DATASET-EXTENSION-MISMATCH( Y | N ) ]
[ WARN-ON-INCOMPLETE-RI( Y | N ) ]
[ WARN-ON-SIMPLE-TABLESPACE( Y | N ) ]

```

COPY command and keyword definitions

Required keywords are described first, followed by optional keywords.

Required COPY keywords

COPY

The COPY command controls all phases of replicating the Db2 table spaces and index spaces that are to be cloned.

- Default: None
- Required: Yes
- Restrictions: None

TARGET-DB2 SSID(*sourcesubsystem*)

DEFVCAT(*targethlq*)

IP(*ppp.ppp.ppp.ppp*)

NAME(*name*)

PORT(*port*)

The TARGET-DB2 parameter specifies the target Db2 subsystem and supplies the parameters needed to access the target via TCP/IP. Also, a default high level qualifier is supplied for the target Db2 data sets.

SSID - the subsystem ID of the target Db2.

DEFVCAT - the default high-level qualifier used for the target data sets if the target Db2 table spaces and index spaces are not pre-defined.

IP - the IP address (IPv4 or IPv6) of the LPAR on which the target Db2 subsystem resides. If NAME or TCP/IP-SERVER-NAME are also specified, those parameters take precedence over IP.

NAME - the DNS name of the LPAR on which the target Db2 subsystem resides. If IP or TCP/IP-SERVER-IP are also specified, NAME takes precedence over those parameters.

PORT - the port that the TCP/IP server uses to wait for requests from the source job and that the source job uses to connect to the TCP/IP server job. The port assigned to the TCP/IP server cannot be a port used by Db2 or any other application.

- Default: None
- Required:
 - SSID is an unconditionally required parm.
 - IP is required if TCP/IP used to connect to the target subsystem.
 - DEFVCAT is required if one or more source table spaces or index spaces do not exist on the target subsystem.
 - PORT is required to use a different port than the one specified in PARMLIB.
- Restrictions: None
- Short form: TDB2

Optional COPY keywords

ALLOW-COPY-ON-MISMATCH (*Y | N*)

The ALLOW-COPY-ON-MISMATCH parameter specifies whether a copy should be allowed if one or more object attributes do not match between the source and target table spaces or index spaces.

N – The table space or index space is not copied and the job completes with a return code of 4.

Y – The copy is still performed if the CKZINI PARMLIB parameter MAX_RC is set to 4 (or if the CKZINI PARMLIB parameter MAX_RC is set to 0 but MAX-RC on the SET command is set to 4).

Note: When this keyword is set to N, set the PARMLIB keywords MAX_COPY_RC to 8 and the MAX_RC to 4 (or use the SET command keywords MAX-COPY-RC(8) and MAX-RC(4)) to allow Db2 Cloning Tool Table Space Cloning to copy only table spaces and index spaces that are safe to copy, such as those not in restrictive or advisory states or those without object mismatches (such as column length or type). For all other source table spaces and index spaces, error messages will be produced and the table spaces and index spaces are not copied.

- Default: N
- Required: No
- Restrictions: None
- Short form: ACOM

ALWAYS-COPY-HISTORY-TABLES (*Y | N*)

ALWAYS-COPY-HISTORY-TABLES can be used to select history tables associated with all objects specified on a LISTDEF. To select both history tables and non-history tables, set ALWAYS-COPY-

HISTORY-TABLES to Y. When ALWAYS-COPY-HISTORY-TABLES is set to N, history tables are not selected; however, history tables can be selected by using the HISTORY parameter on a LISTDEF command.

The following examples show how to combine ALWAYS-COPY-HISTORY-TABLES and the HISTORY parameter on a LISTDEF to select history tables.

Example: Include only non-history tables in a table space:

```
INCLUDE TABLESPACES TABLESPACE table_space_name ALL  
ALWAYS-COPY-INDEXSPACES(N)  
ALWAYS-COPY-HISTORY-TABLES(N)
```

Example: Include only history tables with all history indexes in a table space:

```
INCLUDE TABLESPACES TABLESPACE table_space_name HISTORY ALL  
ALWAYS-COPY-INDEXSPACES(Y)  
ALWAYS-COPY-HISTORY-TABLES(N)
```

Example: Include both history and non-history tables with all indexes in a database.

```
INCLUDE TABLESPACES DATABASE database_name ALL  
ALWAYS-COPY-INDEXSPACES(Y)  
ALWAYS-COPY-HISTORY-TABLES(Y)
```

For more information about how this keyword affects LISTDEF processing, see the topic [“Impact of COPY command keywords on LISTDEF processing”](#) on page 562.

- Default: N
- Required: No
- Restrictions: Valid only for Db2 10 and later subsystems.
- Short form: ACHTB

ALWAYS-COPY-INDEXSPACES (Y | N)

When set to Y, the ALWAYS-COPY-INDEXSPACES parameter specifies that for every table space included in a LISTDEF, all index spaces are also included. No INCLUDE INDEXSPACES syntax is required in the LISTDEF.

For more information about how this keyword affects LISTDEF processing, see the topic [“Impact of COPY command keywords on LISTDEF processing”](#) on page 562.

- Default: Y
- Required: No
- Restrictions: Must be set to Y if RECREATE-OBJECTS is A or M.
- Short form: CPYIX

AUTO-INDEXSPACE-TRANSLATE (Y | N)

When set to Y, the AUTO-INDEXSPACE-TRANSLATE parameter specifies that names of index spaces that were explicitly created will be matched automatically between source and target tables without specifying object translate rules. The only object translation rules that must be specified are for tables.

If AUTO-INDEXSPACE-TRANSLATE is set to Y, Db2 Cloning Tool Table Space Cloning translates the table on which index was created (if needed), searches for the table on the target system, and searches for indexes on the target table to attempt to find a matching index.

It is not recommended to specify index space translation via the OBJECT-TRANSLATE parameter of the COPY command. If AUTO-INDEXSPACE-TRANSLATE is set to Y and translation rules for an index or its index space are specified via the OBJECT-TRANSLATE parameter, the auto-translation algorithm described previously is not used for the index or its index space.

If AUTO-INDEXSPACE-TRANSLATE is set to Y, Db2 Cloning Tool Table Space Cloning always uses CHECK-INDEX-KEYS(Y).



Attention: Use of CHECK-INDEX-KEYS(Y) may cause performance degradation, especially when there are many indexes to copy, or caching is being used to access the catalog (caching is not supported for index keys).

- Default: N
- Required: No
- Restrictions: None
- Short form: AIST

AUTO-START-SOURCE-SPACE (Y | N | R)

The AUTO-START-SOURCE-SPACE parameter allows the source job to optionally start each source Db2 table space and index space after the copy process is complete. This parameter also allows stopped source table space and index spaces that require processing with the UNLOAD utility to be started before utility execution.

Y – (the default) Db2 Cloning Tool Table Space Cloning starts the source table spaces and index spaces in RW mode after the copy is complete, or starts source table spaces and index spaces that are stopped so they can be processed with the UNLOAD utility.

N - source table spaces and index spaces are left stopped after the copy is complete. An additional START command will not be issued for spaces that require processing with the UNLOAD utility.

R– the source table spaces and index spaces are restored to the status they were before executing the source job; however, if a source table space or index space has an initial status of STOP or STOPP, the table space or index space is started in RW. If the space will be processed by the UNLOAD utility, setting this parameter to R is the same as setting the parameter to Y.

If you set this parameter to Y or R, and a table space or index space has a status that is not STOP, RW, RO or STOPP, a RC of 8 is issued and the table space or index space is not copied.

If FUZZY-COPY(Y) has been specified, AUTO-START-SOURCE-SPACE is ignored, except when spaces require processing with the UNLOAD utility.

- Default: Directed by the CKZINI token AUTO_START_SOURCE_SPACE in the PARMLIB :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: Valid only when source table spaces and index spaces have a status of STOP, RW, RO, or STOPP; source table spaces and index spaces with any other status will not be copied by Db2 Cloning Tool Table Space Cloning.
- Short form: ASTRSS

AUTO-START-TARGET-SPACE(Y | N)

The AUTO-START-TARGET-SPACE parameter is passed to the Db2 Cloning Tool Table Space Cloning target job and allows the target job to optionally start each target Db2 table space and index space after the cloning process is complete. This parameter also allows stopped target table space and index spaces that require processing with the LOAD or REBUILD INDEX utility to be started before utility execution. The target space is not started after the source job completes.

Refer to [“How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 190](#) for more information about stopping and starting target table spaces and index spaces.

- Default: Directed by CKZINI token AUTO_START_TARGET_SPACE in the PARMLIB :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: ASTRTS

AUTO-STOP-TARGET-SPACE(Y | N)

The AUTO-STOP-TARGET-SPACE parameter allows the source job to optionally stop the target Db2 table spaces and index spaces. If the target table spaces and index spaces are already stopped, set

this to N. This will prevent the Db2 DISPLAY command from being issued against all the target table spaces and index spaces.

Refer to [“How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 190](#) for more information about stopping and starting target table spaces and index spaces.

- Default: Directed by CKZINI token AUTO_STOP_TARGET_SPACE in the PARM LIB :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: ASTPTS

AUTO-TABLESPACE-TRANSLATE (Y | N)

When set to Y, the AUTO-TABLESPACE-TRANSLATE parameter specifies that names of table spaces that were explicitly created will be matched automatically between source and target tables without specifying object translate rules. The only object translation rules that must be specified are for tables.

If AUTO-TABLESPACE-TRANSLATE is set to Y, Db2 Cloning Tool Table Space Cloning works with the table in the table space. The table is translated if a translation rule for the table was specified. Db2 Cloning Tool Table Space Cloning then searches for the table on the target system. If the table is found, table space data is taken from the target, and source and target table spaces are matched. For LOB and XML table spaces, Db2 Cloning Tool Table Space Cloning works with the table of the base table space of the LOB and XML space. Db2 Cloning Tool Table Space Cloning searches first for the base table on the target, then looks through the Db2 catalog tables to find related LOB and XML tables and their table spaces.

If AUTO-TABLESPACE-TRANSLATE is set to Y and translation rules for a table space are specified via the OBJECT-TRANSLATE parameter, the auto-translation algorithm described previously is not used for the table space.

- Default: N
- Required: No
- Restrictions: None
- Short form: ATST

CATALOG-PREFETCH (ENABLE-PREFETCH (Y | N) ENABLE-SOURCE-PREFETCH (Y | N) ENABLE-TARGET-PREFETCH (Y | N) READ-FROM-SERVER-CACHE (Y | N) TARGET-PREFETCH-DATABASE-LIST (*targetdatabase1, targetdatabase2, . . . targetdatabasesn*))

This parameter can be entered on the COPY command in the source job. For additional information, refer to the topic [“Considerations for using catalog prefetch to populate the object cache” on page 201](#).

- Default: None
- Required: No
- Restrictions: None
- Short form: CATPF

ENABLE-PREFETCH - Y enables the other prefetch commands.

- Default: N
- Required: No
- Restrictions:

When using this command, only specify objects using the LISTDEF statements INCLUDE TABLESPACES DATABASE or INCLUDE TABLESPACES TABLESPACE. In addition, if TABLESPACE object specification is used, it must be a base table space; that is, LOB and XML table spaces cannot be specified. The LOB specification must be ALL; LOB and XML may not be specified. If these restrictions are not acceptable, specify ENABLE-PREFETCH(N) or ENABLE-SOURCE-PREFETCH(N).

- Short form: ENAPF

ENABLE-SOURCE-PREFETCH - Y enables prefetch for the source catalog. This command allows the source objects from one or more databases to be saved in memory (cached) during a single pass of the catalog tables. If this command is set to Y, source databases, table spaces, table partitions, tables, and indexes will be read from the cache. If this command is set to Y in the source job, LISTDEF will be used to find objects to prefetch.

- Default: N
- Required: No
- Restrictions: None
- Short form: ENASPF

ENABLE-TARGET-PREFETCH - Y enables prefetch for the target catalog. This command allows the target objects from one or more databases to be saved in memory (cached) during a single pass of the catalog tables. When Y is specified for this command, the target server cache is cleared and populated. If the cache does not need to be refreshed, objects can be read from the populated cache by using the READ-FROM-SERVER-CACHE command with ENABLE-TARGET-PREFETCH(N). When Y is specified for ENABLE-TARGET-PREFETCH and the TCP/IP server job is in use, the request to populate the target cache and the list of databases comes from the source job.

- Default: N
- Required: No
- Restrictions: Do not specify when target objects do not exist or when using DDL generation.
- Short form: ENATPF

READ-FROM-SERVER-CACHE – Specifies that Db2 Cloning Tool searches for database, table space, table partition, table, and index target objects in the cache, and not in the catalog. When the connection is not a TCP/IP connection, and target prefetch is enabled, these types of objects are always read from the cache. If target prefetch is enabled, this command automatically is set to Y.

- Default: N
- Required: No
- Restrictions: Valid only if REMOTE-CONNECT-TYPE(T)
- Short form: RSERVCH

TARGET-PREFETCH-DATABASE-LIST - Specifies a list of databases to be prefetched when accessing the target catalog. Connection to the target must be TCP/IP. This command can be used in the source job and is passed to the TCP/IP server job if it is in use. More than 1,000 databases can be entered. If this command is not entered, the list of target databases to be cached is generated from the source data set names mapped to target names using object translate. When copying a large number of table spaces, compare source job run times with and without this database list to determine which gives the best performance. When the list of databases is not specified with this command, the list of databases is created from the list of source databases in the LISTDEF, including object translation. If the connection is lost during target server cache populating, the cache is populated only for the objects in the databases that are listed in messages CKZ00302I and CKZ00303I in the target server output.

Important: If all databases are not included in the prefetch database list, the objects will not be found and the source job will not run correctly.

- Default: None.
- Required: No

- Restrictions: Valid only if REMOTE-CONNECT-TYPE(T).
- Short form: TPFDBL

CHECK-DATASET-COMPATIBILITY(Y | N)

CHECK-DATASET-COMPATIBILITY is used to determine whether the VSAM attributes of the source data sets and the target data sets are compatible. CHECK-DATASET-COMPATIBILITY(Y) should be specified only in a first run of a table space cloning source job that is done in simulation mode. First, run the source job with this parameter set to Y and PGM(NONE) (or SIM(A) if PGM(ADRDSSU) is specified). Resolve any data set incompatibilities, then set CHECK-DATASET-COMPATIBILITY to N and run the source job with PGM and SIM set as desired.

To avoid unnecessary CPU and I/O time, do not use CHECK-DATASET-COMPATIBILITY(Y) in a non-simulation run of the source job.

The following attributes are checked:

- If the data set is in extended format.
- If the data set was allocated using extended addressability.
- Whether the data set can be compressed
- Whether the data set is striped.
- Whether the data set can be spanned.

These attributes must be the same between the source and target subsystems. When one or more data set incompatibilities are found, no copies are attempted and the source job ends with RC=8.

Note: If the target objects have been created with DEFINE NO, no comparison is performed.

IDCAMS LISTCAT commands are issued for each data set pair (source and target). IDCAMS commands and responses are output to the CKZLOG data set.

- Default: N.
- Required: No
- Restrictions: None
- Short form: CKDSN

CHECK-INDEX-KEYS(Y | N)

When set to Y, the CHECK-INDEX-KEYS parameter makes additional checks on index compatibility. LIMITKEY is always checked and is normally sufficient. In some cases, a mismatch is missed unless the keys are read and checked also. If using index caching, run once with CHECK-INDEX-KEYS(Y), change any indexes to correct the mismatches, and then run with CHECK-INDEX-KEYS(N) or the default. Use of this command may cause performance degradation, especially when there are many indexes to copy or caching is being used to access the catalog (caching is not supported for index keys).

- Default: N.
- Required: No
- Restrictions: None
- Short form: CIXKY

COPY-IF-NO-DB2-TARGET-OBJECTS (Y | N)

The COPY-IF-NO-DB2-TARGET-OBJECTS parameter specifies if source VSAM data sets are to be copied even if the target Db2 table spaces and index spaces do not exist.

Target objects that were created with DEFINE NO are always copied. Target objects that were created with DEFINE NO and that do not have existing data sets will require additional processing. Refer to [“Considerations for target objects created using DEFINE NO”](#) on page 186 for more information.

- Default: Directed by CKZINI token COPY_IF_NO_DB2_TARGET_OBJECTS in the PARMLIB :DSN_COPY_OPTIONS section. If no token value, then N.
- Required: No

- Restrictions: None
- Short form: CINTO

COPY-IJ-TO-NONEXISTENT-TARGET (Y | N)

Use this keyword to create data sets on the target subsystem without the need to rename and delete temporary data sets on the target. This keyword can be used when the target data sets have not yet been created and the target objects were not created using DEFINE NO; refer to the topic [“Considerations for target objects created using DEFINE NO” on page 186](#). The following statements are correct for objects that were defined using DEFINE YES.

- When target objects exist but do not have data sets (or the data sets are migrated), and COPY-IJ-TO-NONEXISTENT-TARGET is set to N, the target VCAT is taken from the target catalog, data sets are cloned with an F fifth-node qualifier, and IDCAMS DELETE and ALTER statements are generated into the DD that is specified in the parameter IDCAMS-DDN.
- When target objects exist but do not have data sets (or the data sets are migrated) and COPY-IJ-TO-NONEXISTENT-TARGET is set to Y, the target VCAT is taken from the target catalog, data sets are cloned with an I or J fifth-node qualifier that is taken from source catalog, and IDCAMS DELETE and ALTER statements are not generated.
- When target objects do not exist, but the copy parameters allow it (COPY-IF-NO-DB2-TARGET-OBJECTS(Y)), and COPY-IJ-TO-NONEXISTENT-TARGET is set to N, the data sets are cloned with an F fifth-node qualifier, and IDCAMS DELETE and ALTER statements are generated into the DD that is specified in the parameter IDCAMS-DDN. For the target VCAT, Db2 Cloning Tool first attempts to use the OBJECT-TRANSLATE VCAT parameter value. If the parameter was not specified, Db2 Cloning Tool attempts to use the DEFVCAT parameter value. If no DEFVCAT parameter was specified, the target VCAT cannot be set, and the copy is not performed.
- When target objects do not exist, but copy parameters allow it (COPY-IF-NO-DB2-TARGET-OBJECTS(Y)), and COPY-IJ-TO-NONEXISTENT-TARGET is set to Y, the data sets are cloned with an I or J fifth-node qualifier that is taken from the source catalog, and IDCAMS DELETE and ALTER statements are not generated. For the target VCAT, Db2 Cloning Tool first attempts to use the OBJECT-TRANSLATE VCAT parameter value. If the parameter was not specified, Db2 Cloning Tool attempts to use the DEFVCAT parameter value. If there is no DEFVCAT parameter, the target VCAT cannot be set, and the copy is not performed.
- Default: N
- Required: N
- Restrictions: None
- Short form: CITNT

DATA-MASKING (Y | N)

If set to Y, the DATA-MASKING keyword enables data masking.

If you specify DATA-MASKING(Y) for a table, assume that all indexes on that table must be rebuilt. Omit the table's indexes from the LISTDEF, and ensure ALWAYS-COPY-INDEXSPACES is defaulted (N) or specifies N explicitly to prevent all indexes from being included in copy processing. After running the target job, rebuild all the indexes that were omitted from the copy.

If DATA-MASKING is set to Y, Db2 Cloning Tool Table Space Cloning always uses CHECK-INDEX-KEYS(Y).



Attention: Use of CHECK-INDEX-KEYS(Y) may cause performance degradation, especially when there are many indexes to copy, or caching is being used to access the catalog (caching is not supported for index keys).

For additional information about using data masking, see [Chapter 17, “Using data masking with table space cloning,” on page 271](#).

- Default: N
- Required: N
- Restrictions: DATA-MASKING(Y) is not valid with PGM(SRCIMCPY) or PGM(SRCVSCPY).

- Short form: MASKING

```

DATA-MOVER ( PGM( ADRDSSU | EMCAPI | SRCIMCPY | SRCVSCPY | NONE )
[ ,CMDDNAME( ddname ) ]
[ ,DATACLAS( class ) ]
[ ,FASTREP( PREF | REQ | NONE ) ]
[ ,FCTOPPRCPRIARY [ (PRESMIRREQ | PRESMIRPREF | PRESMIRNONE ) ]
[ ,FCTOXRCPRIARY ]
[ ,MGMTCLAS( class ) ]
[ ,NULLSTORCLAS( Y | N ) ]
[ ,REUSE( Y | N ) ]
[ ,STORCLAS( class ) ]
[ ,USE-LAST-CONSISTENT-FLASHCOPY( Y | N ) ]

```

The DATA-MOVER parameter specifies the program to be used to initiate copies and copy options.

Note: If you specify ADRDSSU and are not using job templates, Db2 Cloning Tool adds the VOLCOUNT(ANY) and TGTALLOC(SRC) parameters to the DFSMSdss COPY command. The VOLCOUNT(ANY) parameter results in the allocation of the target data set on as many volumes as required, to a maximum of 59. The TGTALLOC(SRC) parameter allocates the target data set with the same space allocation type as that of the source data set. Using these parameters makes it more likely for DFSMSdss to choose fast replication when copying the source data sets.

PGM(ADRDSSU) is the default; it specifies that COPY is to initiate FlashCopy or SnapShot 'under the covers' via execution of DSS.

PGM(EMCAPI) - species that Db2 Cloning Tool Table Space Cloning is to invoke EMC TimeFinder/Clone to make the copies using the data set snap facility. Refer to the topic [“GLOBAL command values for EMC TimeFinder/Clone Mainframe Snap Facility data set level support”](#) on page 558 for additional information.

PGM(SRCIMCPY) specifies that image copies are to be used as the input to the table space cloning process. When SRCIMCPY is specified, the cloning process identifies the required image copies based on the specified end point (point in time) of the clone. In most cases, the first full image copy that is older than the specified end point is used, and incremental image copies that were taken after the full image copy but before the end point are merged with the full image copy. LOG-APPLY processing is required and is used to apply the logs to bring objects to a consistent state on the target. Before the data sets are copied to the target, pages are fixed (OBIDs are translated and certain page fields are reset). Objects supported are table spaces (except PBR RPN table spaces), index spaces, and LOBs. Image copies can be SHRLEVEL REFERENCE or CHANGE.

Note: SRCVSCPY has been deprecated. Support is likely to be removed in the future. For more information, see [“Deprecated and removed functions and parameters”](#) on page 10.

PGM(SRCVSCPY) specifies that a VSAM-to-VSAM copy process is to be used. This process copies data sets from source to target without using image copies or fast replication. The source objects are read during the copy process, which then performs OBID translation, RBA reset, and log apply when the pages are written to the target data sets. To avoid data inconsistency on the target after cloning, the log apply process is skipped for VSAM-to-VSAM copy. The source objects are only read once. The target job can rebuild indexes and perform table space and index space REPAIRs, as well as restart the target objects. Objects that are supported for cloning using VSAM to VSAM copy are table spaces (except PBR RPN table spaces), index spaces, and LOBs. Indexes can be cloned or they can be rebuilt.

The following prerequisites and restrictions apply for VSAM to VSAM copy:

- Source and target objects must match, or target objects must not exist and must be created by Db2 Cloning Tool Table Space Cloning. It is recommended to resolve all object mismatches before the cloning is run. For the first run of the source job, specify DATA-MOVER PGM(NONE) and EXCLUDE-MISMATCH-PROCESSING(N). Run the source job, and correct the mismatches that are identified in the source job output.
- XML processing and data masking are not supported.

- SKIP-LOG-APPLY(Y) is always used with PGM(SRCVSCPY).
- The VSAM to VSAM copy process cannot use subtasks to multitask multi-data set non-partitioned objects; therefore, SUBTASKS-DATASET-EXTENSIONS is not allowed.
- The following commands are not supported with PGM(SRCVSCPY): SIM(A), SUBTASK-DATASET-EXTENSIONS, and any of the DATA-MOVER subcommands such as USE-LAST-CONSISTENT-FLASHCOPY, FASTREP or FCTOPPRCPRIARY.
- If you are cloning across LPARs, shared DASD between the source and target systems is required.

PGM(NONE) specifies that no DATA-MOVER is to be invoked by COPY. NONE infers that data set copies will be created by you between the execution of the source job and the execution of the target job. When NONE is specified, COPY still captures necessary Db2 catalog information. NONE may also be used to verify object compatibility from source to target and to ensure parameters are correctly specified.

CMDDDDNAME is used only with job templates. The DD name indicates the name of an output data set that contains a job built using the JOB-TEMPLATE parameter of the COPY command. Note that each step in the externally built set of input cards is a separate call to ADRDSSU. The JCL step execution cards are discarded. The DSS input cards consisting of DSS commands such as COPY, INCLUDE, PARALLEL and data set names are sent to ADRDSSU as is. This parameter is not available for PGM(EMCAPI). CMDDDDNAME is not valid with PGM(SRCIMCPY) or PGM(SRCVSCPY).

DATACLAS – Use this keyword to specify the SMS DATACLAS to be used for target data set allocation or reallocation. This keyword is not supported for PGM(NONE) or PGM(ADRDSSU). If this keyword is specified, the DATA-MOVER PGM uses the supplied class for allocating or reallocating target data sets. If the keyword is not specified, the behavior depends on the DATA-MOVER PGM:

- EMCAPI uses SMS defaults.
- SRCIMCPY and SRCVSCPY use the DATACLAS of the existing allocation, if the target data sets exists. If the target data sets do not exist, SMS defaults are used.
- ADRDSSU always allocates or reallocates the target data sets using the DATACLAS of the source data sets.

FASTREP (DSS parameter) - Indicates whether fast replication is preferred (PREF), required (REQ), or not required (NONE). PREF is the default. Db2 Cloning Tool Table Space Cloning will set up the source/target pairs for a fast replication if PREF or REQ is specified. Db2 Cloning Tool Table Space Cloning will allow a 'normal' copy if NONE is specified. If the level of ADRDSSU indicates it supports this keyword, the keyword will be passed to ADRDSSU. FASTREP is not valid with PGM(SRCIMCPY) or PGM(SRCVSCPY), and is ignored for PGM(EMCAPI).

FCTOPPRCPRIARY (DSS parameter) - Indicates that a FlashCopy target volume can also be a PPRC primary volume. This applies to ESS devices only. This does not apply when FASTREP(NONE) is also specified. FCTOPPRCPRIARY is not valid with PGM(SRCIMCPY) or PGM(SRCVSCPY), and is ignored for PGM(EMCAPI). IBM Remote Pair FlashCopy (also known as Preserve Mirror) can be specified by including one of the optional keywords. Preserve Mirror mirrors the FlashCopy command that is issued at the local site to the remote site. This allows FlashCopy operations to occur to PPRC primary volumes without affecting the PPRC duplex state. IBM Remote Pair FlashCopy must be installed in the storage controller along with the corresponding software support in z/OS. In addition, both the source and target volumes being PPRC primary volumes and in the same storage controller and their corresponding PPRC secondary volumes being in the same storage controller. Specify one of the following to use this functionality:

- PRESMIRREQ: Require the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation will not be completed.
- PRESMIRPREF: Prefer the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation is still performed.
- PRESMIRNONE: Do not use Preserve Mirror.

When any of these options is specified, they will be used as part of the source to target volume pairing criteria. The specified option will also be passed to ADRDSSU as part of the generated copy

commands. When FCTOPPRCPRIARY is not specified or if the capability is not supported by the ESS, a PPRC primary volume is not eligible to become a FlashCopy target volume. For additional information about Preserve Mirror, refer to the documentation for your version of z/OS.

Do not specify the FCTOPPRCPRIARY keyword with the FASTREP (NONE) keyword.

FCTOXRCPRIARY (DSS parameter) - Indicates that a FlashCopy target volume can also be an XRC primary device. This keyword is applicable only for DATA-MOVER PGM(ADRDSU). This does not apply when FASTREP(NONE) is also specified.

MGMTCLAS – Use this keyword to specify the SMS MGMTCLAS to be used for target data set allocation or reallocation. This keyword is not supported for PGM(NONE). If this keyword is specified, the DATA-MOVER PGM uses the supplied class for allocating or reallocating target data sets. If the keyword is not specified, the behavior depends on the DATA-MOVER PGM:

- EMCAPI uses SMS defaults.
- SRCIMCPY and SRCVSCPY use the MGMTCLAS of the existing allocation, if the target data sets exists. If the target data sets do not exist, SMS defaults are used.
- ADRDSU uses the MGMTCLAS of the source data sets.

NULLSTORCLAS - This keyword is applicable only for DATA-MOVER PGM(ADRDSU). If set to Y (the default), a null storage class is passed to the automatic class selection (ACS) routine when copying data sets. This allows the ACS routine to assign storage classes to target data sets. If set to N, the input to the ACS will be source data set's storage class. For more information on this keyword, consult the DFSMSdss Storage Administration documentation. This keyword is mutually exclusive with STORCLAS keyword.

REUSE – This keyword is not supported for PGM(NONE) or PGM(ADRDSU). If set to N, the DATA-MOVER PGM reallocates the target data sets during the copy. If set to Y (default), the existing allocation of the target data sets is used. ADRDSU always attempts to reuse the existing allocation.

STORCLAS – Use this keyword to specify the SMS STORCLAS to be used for target data set allocation or reallocation. This keyword is not supported for PGM(NONE) and is mutually exclusive with NULLSTORCLAS keyword. If this keyword is specified, the DATA-MOVER PGM uses the supplied class for allocating or reallocating target data sets. If the keyword is not specified, the behavior depends on the DATA-MOVER PGM:

- EMCAPI uses SMS defaults.
- SRCIMCPY and SRCVSCPY use the STORCLAS of the existing allocation, if the target data sets exists. If the target data sets do not exist, SMS defaults are used.
- ADRDSU uses the STORCLAS of the source data sets.

USE-LAST-CONSISTENT-FLASHCOPY - This keyword allows you to use data sets that have been created by using the Db2 COPY utility with SHRLEVEL CHANGE and FLASHCOPY CONSISTENT options as the source for cloning. This option enables the target objects to be consistent without stopping the source objects and without the need for applying logs. Also, specifying the USE-LAST-CONSISTENT-FLASHCOPY(Y) keyword avoids rebuilding of indexes that are included in a consistent FlashCopy image copy. Using this option along with multiple subtasks can significantly reduce the CPU usage time and elapsed time of cloning. The USE-LAST-CONSISTENT-FLASHCOPY(Y) keyword is not valid with PGM(SRCIMCPY) or PGM(SRCVSCPY). If the USE-LAST-CONSISTENT-FLASHCOPY(Y) keyword is specified, then the UNLOAD-LOAD and LOG-APPLY features are automatically switched off.

Note: Specifying this option requires that the version and the object attributes of the source objects are identical to the objects in the consistent FlashCopy image copy.

DATASETS-TO-COPY-DDN(*ddname*)

The DATASETS-TO-COPY-DDN parameter supplies the *ddname* pointing at the data set which will contain a list of TO and FROM data set names derived from the LISTDEF command input. The data set pointed to by DATASETS-TO-COPY-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in replicating the Db2 VSAM objects when using a method other than Db2 Cloning Tool Table Space Cloning.

The data sets to copy and their names are output in CKZPRINT also. This is just a more convenient form for submitting copy commands to a copy product that Db2 Cloning Tool Table Space Cloning does not currently support. Refer to the JOB-TEMPLATE parameter for the preferred method of building copy jobs to be run outside of Db2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No. This is optional even if PGM(NONE).
- Restrictions: Valid only if DATA-MOVER PGM(NONE) is specified.
- Short form: DSN-DDN

DSNS-PER-COPY(*numberofdatasets*)

The DSNS-PER-COPY parameter specifies the number of data sets to send to DSS in a single copy command. Valid values are 1 to 255; 255 is the recommended value.

- Default: PARMLIB specification for DSNS_PER_COPY, or 255 if not specified in PARMLIB.
- Required: No.
- Restrictions: None
- Short form: DSNSPC

DSS-COPY-COMMANDS (*numberofcommands*)

The DSS-COPY-COMMANDS parameter specifies the number of DSS copy commands to send to DSS in a single invocation. Valid values are 1 to 256; specifying a large value may result in a storage shortage and S878 abends. When cloning a large number of data sets, the default of 24 might be too high.

- Default: PARMLIB specification for DSS_COPY_COMMANDS, or 5 if not specified in PARMLIB.
- Required: No
- Restrictions: None
- Short form: DSSCMDS

ENCRYPTION-MISMATCH-RC (0 | 4 | 8)

Use this parameter to control encryption-related mismatch processing. The following encryption-related mismatches are checked:

- If the combination of encryption statuses of the source and target data sets is supported for the selected data mover.
- If the combination of encryption statuses of the source and target data sets might result in an error when copying the data sets outside of Db2 Cloning Tool.
- If the copy will result in reallocating an encrypted target data set as non-encrypted.
- If an encrypted target data set will be usable with the key label definitions of the target LPAR.

When one of the listed mismatches is detected, the specified value of ENCRYPTION-MISMATCH-RC controls further processing, as follows:

- Return code 0 issues an informational message and is treated as if there is no mismatch.
- Return code 4 issues a warning message. If UNLOAD-LOAD is enabled, it will be used to resolve the mismatch.
- Return code 8 issues an error message. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH.

For more information about copying encrypted objects, see [“Considerations for DFSMS pervasive encryption” on page 187.](#)

- Default: 4
- Required: No
- Restrictions: None
- Short form: ENCMMRC

EXCLUDE-MISMATCH-PROCESSING(Y | N)

When this parameter is Y, no mismatch checking is performed. However, before specifying EXCLUDE-MISMATCH-PROCESSING (Y), you might want to ensure that mismatches have been identified and corrected prior to running the source job.

When EXCLUDE-MISMATCH-PROCESSING is set to Y, all objects are copied regardless of object mismatches and regardless of the setting for ALLOW-COPY-ON-MISMATCH. When EXCLUDE-MISMATCH-PROCESSING (N) is specified, ALLOW-COPY-ON-MISMATCH controls whether mismatches are copied.

To ensure there are no mismatches that might result in an unsuccessful copy, it is highly recommended that you run the source job twice, as follows:

1. For the first run of the source job, specify DATA-MOVER PGM(NONE) and EXCLUDE-MISMATCH-PROCESSING(N). Run the source job, and correct the mismatches that are identified in the source job output. Rerun the source job until no mismatches are found.
2. For the second run, copy the first source job into a separate member. Modify the new member to specify EXCLUDE-MISMATCH-PROCESSING(Y) and set DATA-MOVER PGM to the desired copy method (such as ADRDSSU).

- Default: N
- Required: No
- Restrictions: None
- Short form: EXMMP

FUZZY-COPY(Y | N)

The FUZZY-COPY parameter disallows Db2 Cloning Tool Table Space Cloning from stopping the source table spaces and index spaces before replicating them.

Refer to [“How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 190](#) for more information about stopping and starting table spaces and index spaces.

If Y, ADRDSSU is invoked by Db2 Cloning Tool Table Space Cloning with TOLERATE(ENQFAILURE) or the EMC API is invoked with TOLERATEENQFAILURE(Y). If you specify Y, note that RACF FACILITY class authority to process data sets even though shared or exclusive access fails.

If Y, the VSAM-to-VSAM process always uses SKIP-LOG-APPLY(Y), because specifying DATA-MOVER PGM(SRCVSCP) and FUZZY-COPY(Y) might cause data inconsistency on the target.



Attention: For DATA-MOVER PGM(ADRDSSU) or DATA-MOVER PGM(EMCAPI), specifying FUZZY-COPY(Y) might create inconsistent data unless you also specify LOG-APPLY. Refer to [“Considerations for using FUZZY-COPY” on page 201](#) for additional information about using this parameter.

- Default: N
- Required: No
- Restrictions: Valid only if DATA-MOVER PGM(ADRDSSU), DATA-MOVER PGM(EMCAPI), or DATA-MOVER PGM(SRCVSCP) is specified.
- Short form: FUZZY

IDCAMS-DDN(*ddname*)

The IDCAMS-DDN parameter specifies the name of a data set where Db2 Cloning Tool Table Space Cloning will write out IDCAMS commands to delete an object data set and rename the copied F0001 data set to the object data set name. The data set pointed to by IDCAMS-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in replicating the Db2 VSAM objects when the Db2 table spaces and index spaces do not already exist on the target, either because the target object does not exist or was created with the DEFINE NO attribute.

- Default: None
- Required: No

- Restrictions: Valid when the table spaces or index spaces do not exist in the target subsystem.
- Short form: IDC-DDN

IGNORE-RF-MISMATCH-IF-NO-VAR-COLS(Y | N)

This parameter can be used to allow copy of table spaces from source to target when there is a mismatch involving reordered row format. For example, the source may have been migrated from Db2 V8 and be in basic row format (BRF), and the target objects may have been created on Db2 V9 NFM and be in reordered row format (RRF). When IGNORE-RF-MISMATCH-IF-NO-VAR-COLS is set to Y, table spaces with no variable columns can ignore the mismatch in row format and be copied without a warning. Variable columns are VARCHAR, LONGVAR, VARG, LONGVARG, and VARBIN.

When IGNORE-RF-MISMATCH-IF-NO-VAR-COLS is N, all reordered row mismatches are treated as mismatches. A copy of mismatched spaces is only allowed when ALLOW-COPY-ON-MISMATCH(Y). If ALLOW-COPY-ON-MISMATCH(Y) and IGNORE-RF-MISMATCH-IF-NO-VAR-COLS(Y), and there are one or more variable columns, a warning is issued and the space is copied. Note that any other incompatibility will override the processing of this command and ALLOW-COPY-ON-MISMATCH will determine if the space is copied.

- Default: None
- Required: No
- Restrictions: None
- Short form: VARCL

INCLUDE-ALL-RI (Y | N)

If set to Y, specifies that RI is to be added to all LISTDEFS. The RI indicator is automatically inserted in all LISTDEF statements.

For more information about how this keyword affects LISTDEF processing, see the topic [“Impact of COPY command keywords on LISTDEF processing”](#) on page 562.

- Default: N. If DDL-ENABLE(Y) and PROCESS-TYPE is Y, A, or G, the default is Y. If DATA-MASKING(Y), the default is Y.
- Required: No
- Restrictions: Must be set to Y if RECREATE-OBJECTS is A or M.
- Short form: INCRI

JOB-TEMPLATE (*inddname1, outddname1,...[, inddnamen, outddnamen]*)

The JOB-TEMPLATE parameter passes in DD name pairs of job templates. Up to 100 DD name pairs can be specified. These DD name pairs are processed independently of the PGM value (ADRDSSU, EMCAP1, or NONE) specified in the DATA-MOVER COPY subcommand. Non-SMS volume specification must be done manually when using job templates. See the sample member CKZJOB1 in the JCL library.

- Default: None.
- Required: N
- Restrictions: None
- Short form: JTEMP

```

LOG-APPLY ( LA-ENABLE( Y | N )
  [ ,ALLOW-SKIP-INACTIVE-MEMBERS( Y | N ) ]
  [ ,COMMON-CONSISTENT-POINT( Y | N ) ]
  [ ,DATA-SHARING-MEMBERS( id | ssid | zparm ) ]
  [ ,DELETE-MINILOG-DATASETS( Y | N ) ]
  [ ,END-POINT( TO CURRENT | TO LOGPOINT X'byte_string' | TO QUIESCE | TO_TIMESTAMP
timestamp ) ]
  [ ,GMT-OFFSET( +hh:mm | -hh:mm ) ]
  [ ,IMAGE-COPY-PREFERENCE( copy_types ) ]
  [ ,INDEX-LOG-APPLY ( Y | N ) ]
  [ ,MINILOG-HLQ( miniloghighlevelqualifier ) ]
  [ ,MINILOG-LARGE-FILE-TYPE( Y | N ) ]
  [ ,MINILOG-UNIT-TYPE( SYSALLDA ) ]
  [ ,MINILOG-QUANTITY-IN-TRACKS( Y | N ) ]
  [ ,MINILOG-PRIMARY-QUANTITY( nnnn ) ]
  [ ,MINILOG-SECONDARY-QUANTITY( nnnn ) ]
  [ ,MINILOG-VOLUME-COUNT( nnn ) ]
  [ ,MINILOG-DATACLAS( dataclas ) ]
  [ ,MINILOG-STORCLAS( storclas ) ]
  [ ,MINILOG-MGMTCLAS( mgmtclas ) ]
  [ ,NUMBER-OF-BUFFERS( nn ) ]
  [ ,NUMBER-OF-CHANNEL-PROGRAMS( nn ) ]
  [ ,NUMBER-OF-SORTS( nn ) ]
  [ ,QUIESCE-POINT( Y | N ) ]
  [ ,SKIP-LOG-APPLY ( Y | N ) ]
  [ ,SKIP-SYSLGRNX( Y | N ) ]
  [ ,SORTFILE-LARGE-FILE-TYPE( Y | N ) ]
  [ ,SORTFILE-UNIT-TYPE( SYSALLDA ) ]
  [ ,SORTFILE-PRIMARY-QUANTITY( nnnn ) ]
  [ ,SORTFILE-SECONDARY-QUANTITY( nnnn ) ]
  [ ,SORTFILE-VOLUME-COUNT( nnn ) ]
  [ ,SORTFILE-DATACLAS( dataclas ) ]
  [ ,SORTFILE-STORCLAS( storclas ) ]
  [ ,SORTFILE-MGMTCLAS( mgmtclas ) ]
  [ ,SORT-PROGRAM( DFSORT | DB2SORT | SYNCSORT ) ]
  [ ,SPACES-PER-MINILOG( numberofspaces ) ]
  [ ,SPACES-PER-LOG-APPLY-CALL( numberofspacespercall ) ]
  [ ,UNIFIED-WARNING( Y | N ) ]
  [ ,USE-QUIESCE-POINT-FOR-LOGPOINT( Y | N ) ]
  [ ,USE-TCPIP( Y | N ) ]
  [ ,WARN-IF-SKIP-QUIESCE( Y | N ) ]
  [ ,WARN-IF-TS-DEFINED-LOG-NO( Y | N ) ]
  [ ,WORKFILE-LARGE-FILE-TYPE( Y | N ) ]
  [ ,WORKFILE-UNIT-TYPE( SYSALLDA ) ]
  [ ,WORKFILE-QUANTITY-IN-TRACKS( Y | N ) ]
  [ ,WORKFILE-PRIMARY-QUANTITY( nnnn ) ]
  [ ,WORKFILE-SECONDARY-QUANTITY( nnnn ) ]
  [ ,WORKFILE-VOLUME-COUNT( nnn ) ]
  [ ,WORKFILE-DATACLAS( dataclas ) ]
  [ ,WORKFILE-STORCLAS( storclas ) ]
  [ ,WORKFILE-MGMTCLAS( mgmtclas ) ]
  [ ,ZPARM-MEMBER( zparmmembername ) ] )

```

LOG-APPLY is used in the source job and allows log records written by Db2 from before the copies in the source job until the target job is run to be applied to Db2 pages being updated in the target job.

LOG-APPLY is required when image copies are used as input to the cloning process; it is used to apply logs to the image copies and bring the cloned objects to the desired consistent point in time. Additional parameters can be used to specify the consistent point of the cloning process.

- Default: None
- Required: Yes, if PGM (SRCIMCPY) or PGM(SRCVSCPY) is specified.
- Restrictions: None
- Short form: LOGAP

LA-ENABLE- Set this parameter to Y to enable log apply. If this parameter is set to Y, you must include the ZPARAM-MEMBER parameter. If LA-ENABLE is Y and the source objects are in a data sharing group, you must specify the DATA-SHARING-MEMBERS parameter. If LA-ENABLE is N, the parameters are validated but no log changes will be applied to the target.

- Default: N
- Required: Yes, if PGM (SRCIMCPY) or PGM(SRCVSCPY) is specified.
- Restrictions: LA-ENABLE(Y) cannot be used for cloning PBR RPN table spaces.
- Short form: LAENA

ALLOW-SKIP-INACTIVE-MEMBERS - This parameter can be used for cloning in environments where some data sharing members have not been used for a long time and may have had their ZPARAMs deleted, and therefore do not need to be considered during log read. Set this parameter to Y to exclude log apply processing for those members that are not specified in the DATA-SHARING-MEMBERS keyword. Inactive members should be deleted from the Db2 control file or deleted manually from the DATA-SHARING-MEMBERS keyword in the generated source job. The source job still verifies that all active data sharing group members are included in the DATA-SHARING-MEMBERS keyword; skipping active members will result in an error message. For additional information, see [“LOG-APPLY in data sharing environments when one or more members are not available” on page 295.](#)

- Default: N
- Required: No
- Restrictions: None
- Short form: ASKIM

COMMON-CONSISTENT-POINT - Set this parameter to Y to have all table space objects brought to the same consistent point. For PGM(SRCIMCPY), if no common consistent point can be found, the spaces are not processed (unless UNIFIED-WARNING (Y) is specified).

- Default: N
- Required: No
- Restrictions: None
- Short form: COMCP

DATA-SHARING-MEMBERS - When the source subsystem is a data sharing group, identify the members of the group using this parameter. Specify an identification number, subsystem ID, and ZPARAM member for each member in the data sharing group. For example:

```
DATA-SHARING-MEMBERS (
  1, SS1A, SS1APARM
  2, SS1B, SS1BPARM )
```

For additional information, see [“LOG-APPLY in data sharing environments when one or more members are not available” on page 295.](#)

- Default: None
- Required: Required if the source subsystem is a data sharing group
- Restrictions: None
- Short form: DSMBR

DELETE-MINILOG-DATASETS - If set to YES, the minilog data sets that are generated by the source TCP/IP server job will be deleted while the target job is running.

- Default: N
- Required: No
- Restrictions: Valid only for cross-LPAR cloning if LOG-APPLY is enabled and REMOTE-CONNECT-TYPE (T) is specified, or if REMOTE-CONNECT-TYPE is not specified
- Short form: MLDEL

END-POINT - This parameter specifies when Db2 Cloning Tool Table Space Cloning is to stop looking for a consistent set of objects. This parameter is required for PGM(SRCIMCPY). Specify one of the following:

- TO_CURRENT: (Default) Current point in time.
- TO_LOGPOINT X'*byte-string*': The specified log point as 12 or 20 hexadecimal digits. If DSNJCNVT is set and 10-byte RBAs are in use, all byte strings must be 10 bytes. When this value is specified, logs are applied up to this specific log point.

Note: The USE-QUIESCE-POINT-FOR-LOGPOINT parameter results in the source job building the END-POINT(TO_LOGPOINT) command using the quiesce point that is generated from the QUIESCE utility.

- TO_QUIESCE: Last (most recent) quiesce point.
- TO_TIMESTAMP *timestamp*: The specified log point in Db2 TIMESTAMP format. Specify a timestamp that is within the boundaries of the logs that are recorded in the BSDS. This directs log apply processing to read the log and incorporate data up to the specified timestamp.
- Default: END-POINT(TO_CURRENT)
- Required: No, unless PGM(SRCIMCPY) is specified
- Restrictions: None
- Short form: ENDPT

GMT-OFFSET - This parameter can be used to set a specific GMT offset for TIMESTAMP conversion to LRSN/RBA values, instead of allowing the cloning process to derive the offset from the system. This might be necessary if you want to select image copies during the cloning process with timestamp values from SYSIBM.SYSCOPY that were created before or after a daylight savings time change occurred. To use this parameter, specify USE-LOCAL-TIME(N) and specify the GMT offset in the format *+hh:mm* or *-hh:mm*, where *hh* is hours and *mm* is minutes. Value values are from -12:00 to +14:00.

- Default: None
- Required: No
- Restrictions: None
- Short form: GMT

IMAGE-COPY-PREFERENCE - This parameter is used only for PGM(SRCIMCPY); it is ignored when using target job log apply. It allows you to restrict the types of image copies that are processed by log apply and the order in which they are processed. The input to this parameter is one to five pairs of alphabetic characters that indicate a type of image copy. If you omit a pair, that type of image copy is not processed. Valid character pairs are:

- FC: FlashCopy
- LP: local primary
- LB: local backup
- RP: recovery primary
- RB: recovery backup

For example, if you specify IMAGE-COPY-PREFERENCE (LPLBRPRB), the order in which image copies are selected is LP, LB, RP, and RB. In addition, FlashCopy image copies are not selected for log apply processing.

- Default: FCLPLBRPRB

- Required: No
- Restrictions: Specify only in the source job
- Short form: IMCPP

INDEX-LOG-APPLY - This parameter can be used only when cloning from source data sets. If set to Yes, Db2 Cloning Tool Table Space Cloning uses the log apply process on indexes when possible, instead of rebuilding them. The log apply process is performed after the index spaces are cloned to the target Db2 subsystem. If INDEX-LOG-APPLY is set to Y, then SKIP-SYSLGRNX is automatically set to N to prevent potential issues with indexes after the cloning.

- Default: No
- Required: No
- Restrictions: Only valid when PGM(ADRDSSU), PGM(EMCAPI), or PGM(NONE) is specified
- Short form: IXLA

MINILOG-HLQ - Specify the high level qualifier for the minilog data sets. Up to 35 characters are permitted; the last qualifier is automatically generated.

- Default: CKZ.MINILOG
- Required: No
- Restrictions: None.
- Short form: MLHLQ

MINILOG-LARGE-FILE-TYPE - Specify Y to indicate that dynamic allocation of the minilog data set should include the LARGE attribute. This allows for data sets to exceed 65,535 tracks.

- Default: N
- Required: No
- Restrictions: None
- Short form: MLTYPE

MINILOG-UNIT-TYPE - Specify the unit type for the minilog data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: MLUNIT

MINILOG-QUANTITY-IN-TRACKS - Specify Y if the minilog is to be allocated in tracks or N if the minilog is to be allocated in cylinders.

- Default: N
- Required: No
- Restrictions: None
- Short form: MLTRK

MINILOG-PRIMARY-QUANTITY - Specify the minilog data set's primary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: MLPQT

MINILOG-SECONDARY-QUANTITY - Specify the minilog data set's secondary quantity.

- Default: 250
- Required: No

- Restrictions: None
- Short form: MLSQT

MINILOG-VOLUME-COUNT - Specify the maximum number of volumes that the minilog data set will require. If SMS parameters are specified to allow for multi-volume DASD data sets, this parameter must be at least as large as the number of volumes that the minilog will ultimately occupy. Enter a value between 1-255 inclusive, or blank to omit the volume count parameter.

- Default: blank
- Required: No
- Restrictions: None
- Short form: MLVOL

MINILOG-DATACLAS - If the minilog data set will be managed by SMS, specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: MLDATA

MINILOG-STORCLAS - If the minilog data set will be managed by SMS, specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: MLSTOR

MINILOG-MGMTCLAS - If the minilog data set will be managed by SMS, specify the SMS Management Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: MLMGMT

NUMBER-OF-BUFFERS - If you want to adjust the BUFNO parameter in JCL, you can enter this parameter as a QSAM override.

- Default: 5
- Required: No
- Restrictions: None
- Short form: NRBUF

NUMBER-OF-CHANNEL-PROGRAMS - If you want to adjust the NCP parameter (the number of read or write macro instructions) in JCL, use this parameter as a QSAM override.

- Default: 1
- Required: No
- Restrictions: None
- Short form: NRCHP

NUMBER-OF-SORTS - This parameter controls the number of sorts that Db2 Cloning Tool Table Space Cloning runs concurrently. Valid values are 1 - 99. This command can reduce or eliminate storage ABENDs resulting from excessive sorts that are running concurrently.

- Default: 1
- Required: No
- Restrictions: Specify only in the source job

- Short form: NRSRT

QUIESCE-POINT - If this parameter is set to Y, a QUIESCE is issued after copies are complete. This parameter can be used with USE-QUIESCE-POINT-FOR-LOGPOINT to allow the source job to build an END-POINT(TO_LOGPOINT X'*byte-string*') parameter, where X'*byte-string*' is the RBA or LRSN from the QUIESCE utility that is submitted by the source job. Both USE-QUIESCE-POINT-FOR-LOGPOINT and QUIESCE-POINT must be set to Y for this function to be invoked.

- Default: N
- Required: No
- Restrictions: None
- Short form: QUIES

SKIP-LOG-APPLY - This parameter can be used to determine whether Db2 Cloning Tool Table Space Cloning will skip log read and log apply after an image copy has been selected for cloning from source image copies. This might be necessary if you want the same data consistency as the selected image copies. SKIP-LOG-APPLY(Y) is always used with DATA-MOVER PGM(SRCVSCPY) and FUZZY-COPY(Y).



Attention: Using SKIP-LOG-APPLY(Y) and END-POINT(TO_CURRENT) ensures consistency in target Db2 objects only when all cloned source objects have consistent FlashCopy image copies that were created in one call of the COPY utility, and when these image copies are the last image copies for the cloned objects.

- Default: N
- Required: No
- Restrictions: Only valid when DATA-MOVER PGM(SRCIMCPY) or DATA-MOVER PGM(SRCVSCPY) is specified. Automatically set to Y when DATA-MOVER PGM(SRCVSCPY) and FUZZY-COPY(Y) is specified.
- Short form: SKIP-LA

SKIP-SYSLGRNX - This parameter can be used to skip SYSLGRNX processing during log read. The recommended value for this parameter is N, unless there are performance problems during log read that may be related to a large number of records in SYSIBM.SYSLGRNX. In this case, setting this parameter to Y may improve performance. If INDEX-LOG-APPLY is set to Y, then SKIP-SYSLGRNX is automatically set to N to prevent potential issues with indexes after the cloning.

- Default: N
- Required: No
- Restrictions: Automatically set to N when INDEX-LOG-APPLY is set Y.
- Short form: SKSLGRX

SORTFILE-LARGE-FILE-TYPE - Specify Y to indicate that dynamic allocation of the sort file data set should include the LARGE attribute. This allows for data sets to exceed 65,535 tracks.

- Default: N
- Required: No
- Restrictions: None
- Short form: SFTYPE

SORTFILE-UNIT-TYPE - Specify the unit type for the sort file data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: SFUNIT

SORTFILE-QUANTITY-IN-TRACKS - Specify Y if the sort file is to be allocated in tracks or N if the sort file is to be allocated in cylinders.

- Default: N
- Required: No
- Restrictions: None
- Short form: SFTRK

SORTFILE-PRIMARY-QUANTITY - Specify the sort file data set's primary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: SFPQT

SORTFILE-SECONDARY-QUANTITY - Specify the sort file data set's secondary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: SFSQT

SORTFILE-VOLUME-COUNT - Specify the maximum number of volumes that the sort file data set will require. If SMS parameters are specified to allow for multi-volume DASD data sets, this parameter must be at least as large as the number of volumes that the sort file will ultimately occupy. Enter a value between 1-255 inclusive, or blank to omit the volume count parameter.

- Default: blank
- Required: No
- Restrictions: None
- Short form: SFVOL

SORTFILE-DATACLAS - If the sort file data set will be managed by SMS, specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: SFDATA

SORTFILE-STORCLAS - If the sort file data set will be managed by SMS, specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: SFSTOR

SORTFILE-MGMTCLAS - If the sort file data set will be managed by SMS, specify the SMS Management Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: SFMGMT

SORT-PROGRAM - Specify the sort program to be used. Valid values are DFSORT, DB2SORT, or SYNCSORT. When DB2SORT is specified, the recommended value of SPACES-PER-MINILOG should be greater than 36 to avoid problems with CSA space constraints.

- Default: DFSORT

- Required: No
- Restrictions: None
- Short form: SORTP

SPACES-PER-MINILOG - Specify the number of spaces included in each minilog data set created by log apply. Db2 Cloning Tool Table Space Cloning attempts to create no more than 36 minilogs, but this is a theoretical maximum and can be impacted by virtual storage constraints. If a value for SPACES-PER-MINILOG is small enough that more than 36 minilog data sets would be created, this parameter value will be raised. For example, if 1,000 base table spaces are to be processed and SPACES-PER-MINILOG is set to 25, 40 minilog data sets would be required. Db2 Cloning Tool Table Space Cloning will raise the SPACES-PER-MINILOG value to allow 36 or less minilog data sets to be create. In this example, 25 would be changed to 28, thereby creating 36 minilog data sets.

If SORT-PROGRAM (DB2SORT) is specified, the recommended value of SPACES-PER-MINILOG should be greater than 36 to avoid problems with CSA space constraints.

- Default: 010
- Required: No
- Restrictions: None
- Short form: SPPML

SPACES-PER-LOG-APPLY-CALL - Specify the number of spaces to be included in each call of the log apply module when PGM(ADRDSU), PGM(EMCAPI), or PGM(NONE) is specified and when remote connection type TCPIP is used for cross-LPAR cloning. Valid values are 1-3000. If the value of SPACES-PER-LOG-APPLY-CALL is specified but out of range, then SPACES-PER-LOG-APPLY-CALL(3000) is used and warning message CKZ535117W is displayed.

If SPACES-PER-LOG-APPLY-CALL is specified, then Db2 Cloning Tool Table Space Cloning separates cloned spaces into groups with no more than the specified SPACES-PER-LOG-APPLY-CALL value in each group, and no more than one group of spaces is processed in each call of the log apply module. If the SPACES-PER-LOG-APPLY-CALL keyword is not specified in the source job, then Db2 Cloning Tool Table Space Cloning processes all cloned spaces in one call of the log apply module.

If you encounter storage shortage and S878 abends in the target job for cross-LPAR cloning scenarios with remote connection type TCPIP and LA-ENABLE(Y), and DATA-MOVER PGM(ADRDSU), DATA-MOVER PGM(EMCAPI), or DATA-MOVER PGM(NONE), then use the SPACES-PER-LOG-APPLY-CALL parameter to separate cloned spaces into groups for each call of the log apply module. The recommended initial value of SPACES-PER-LOG-APPLY-CALL is 3000.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER PGM(ADRDSU), DATA-MOVER PGM(EMCAPI), or DATA-MOVER PGM(NONE) is specified and remote connection type TCPIP is used.
- Short form: SPPLAC

UNIFIED-WARNING - Set this parameter to Y to process commands even if some objects cannot be copied to the specified end point. If this parameter is set to Y with PGM(SRCIMCPY), and COMMON-CONSISTENT-POINT is set to Y, processing continues with a return code of 4 and warning messages are generated. If this parameter is set to N and no common consistent point is found, processing is halted and error messages are generated.

- Default: N
- Required: No
- Restrictions: None.
- Short form: UNWRN

USE-LOCAL-TIME – If this parameter is set to Y, the local time is used for the value in the TO-TIMESTAMP field.

- Default: N
- Required: No
- Restrictions: None
- Short form: LOCAL

USE-QUIESCE-POINT-FOR-LOGPOINT - This parameter is used with QUIESCE-POINT(Y) to automatically generate an END-POINT(TO_LOGPOINT X'*byte-string*') parameter in the source job, where *byte-string* is the quiesce point RBA or LRSN from the QUIESCE utility that is submitted by the source job. You can use this parameter to run the target job at a later time without having to remember the RBA or LRSN from the QUIESCE. Both USE-QUIESCE-POINT-FOR-LOGPOINT and QUIESCE-POINT must be set to Y for this function to be invoked.

- Default: N
- Required: No
- Restrictions: Specify only in the source job
- Short form: UQPFL

USE-TCPIP(Y|N) - This parameter specifies whether to use TCP/IP to communicate with the source Db2 subsystem when cloning between Db2s on two different LPARs This parameter is required for cross LPAR log apply.

- Default: N
- Required: No
- Restrictions: Specify in the SOURCE job
- Short form: USETCP

WARN-IF-SKIP-QUIESCE - If this parameter is set to Y, a warning message is issued if QUIESCE must be skipped for a status that prevents a QUIESCE point. The following states prevent a QUIESCE from being started for the table space(s) being copied: auxiliary CHECK-pending (ACHKP), CHECK-pending (CHKP), COPY-pending (COPY), REBUILD-pending (RBDP), and RECOVER-pending (RECP).

- Default: N
- Required: No
- Restrictions: None
- Short form: WIFSQ

WARN-IF-TS-DEFINED-LOG-NO - This parameter is used only for PGM(SRCIMCPY) and PGM(SRCVSCPYP); it is ignored when using target job log apply. When a base or LOB table space has the NOT LOGGED attribute, Db2 does not create logs for the space. This could result in errors when the source object is copied to the target using PGM(SRCIMCPY) or PGM(SRCVSCPYP). If WARN-IF-TS-DEFINED-LOG-NO is set to Y, a warning message is output for each table space with the NOT LOGGED attribute. If WARN-IF-TS-DEFINED-LOG-NO is set to N, an informational message is output for each table space with the NOT LOGGED attribute.

- Default: N
- Required: No
- Restrictions: None
- Short form: WXLOG

WORKFILE-LARGE-FILE-TYPE - Specify Y to indicate that dynamic allocation of the work file data set should include the LARGE attribute. This allows for data sets to exceed 65,535 tracks.

- Default: N
- Required: No
- Restrictions: None
- Short form: WFTYPE

WORKFILE-UNIT-TYPE - Specify the unit type for the work file data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: WFUNIT

WORKFILE-QUANTITY-IN-TRACKS - Specify Y if the work file is to be allocated in tracks or N if the work file is to be allocated in cylinders.

- Default: N
- Required: No
- Restrictions: None
- Short form: WFTRK

WORKFILE-PRIMARY-QUANTITY - Specify the work file data set's primary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: WFPQT

WORKFILE-SECONDARY-QUANTITY - Specify the work file data set's secondary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: WFSQT

WORKFILE-VOLUME-COUNT - Specify the maximum number of volumes that the work file data set will require. If SMS parameters are specified to allow for multi-volume DASD data sets, this parameter must be at least as large as the number of volumes that the sort file will ultimately occupy. Enter a value between 1-255 inclusive, or blank to omit the volume count parameter.

- Default: blank
- Required: No
- Restrictions: None
- Short form: WFSQT

WORKFILE-DATACLAS - If the work file data set will be managed by SMS, specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: WFDATA

WORKFILE-STORCLAS - If the work file data set will be managed by SMS, specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: WFSTOR

WORKFILE-MGMTCLAS - If the work file data set will be managed by SMS, specify the SMS Management Class.

- Default: None

- Required: No
- Restrictions: None
- Short form: WFMGMT

ZPARAM-MEMBER - This parameter specifies the ZPARAM member name for the source subsystem. This parameter is required when LA-ENABLE is Y.

- Default: None
- Required: Required if LA-ENABLE =Y
- Restrictions: None
- Short form: ZPARAM

LONGVAR-COMPATIBILITY(Y | N)

This keyword prevents a mismatch message when running a source job in which the source objects have LONGVAR columns and the target objects have corresponding VARCHAR columns (or vice versa). This situation may occur if objects have been migrated to Db2 Version 9.1. After migration, existing LONGVAR columns remain as LONGVAR; however, new LONGVAR columns become VARCHAR columns. When these objects are contained in a source job, Db2 Cloning Tool Table Space Cloning will issue a mismatch warning. If you specify Y for LONGVAR-COMPATIBILITY, the mismatch message and its corresponding return code of 4 are suppressed.

Note: The lengths of the corresponding columns must be the same; if they are not, data may be truncated or a Db2 abend may occur.

- Default: Y
- Required: No
- Restrictions: None
- Short form: LONGV

OBJECT-MISMATCH-RETURN-CODE(*mismatchtype1,returncode1, . . . mismatchtypen,returncoden*)

This keyword allows you to change the return code associated with a particular object mismatch. Mismatches are detected after the source and target objects are identified and before the copy starts.

Individual mismatches may be assigned a return code of 0, 4, or 8. If the return code is 0, an informational message is issued and the object is treated as if there is no mismatch. If the return code is 4, warning message is issued. If the return code is 8, an error message is issued and no copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The default return code is 4 if the mismatch return code command is not specified.

mismatchtype must be one of the following:

- CL_CACHE
- CL_COLNO
- CL_COLTYPE
- CL_CYCLE
- CL_DEFAULT
- CL_FLDPROC
- CL_HASHKEY_COLSEQ
- CL_INCREMENT
- CL_LENGTH
- CL_MAXVALUE
- CL_MINVALUE
- CL_SCALE
- CL_SEQTYPE
- CL_SOURCETYPEID

- CL_START
- IP_DSSIZE
- IP_LIMITKEY
- IX_CLUSTERING
- IX_COLCOUNT
- IX_COLNO
- IX_COLSEQ
- IX_COMPRESS
- IX_DSSIZE
- IX_HASH
- IX_INDEXTYPE
- IX_NR_PARTS
- IX_ORDERING
- IX_PADDED
- IX_PAGENUM
- IX_PGSIZE
- IX_PIECESIZE
- IX_SPARSE
- IX_UNIQUERULE
- IX_UNIQUE_COUNT
- TB_COLCOUNT
- TB_EDPROC
- TB_HASHKEYCOLUMNS
- TB_VERSION
- TP_COMPRESS
- TP_DSSIZE
- TP_FORMAT
- TP_LIMITKEY
- TS_CCSD
- TS_CLONE
- TS_COMPRESS
- TS_COMPRESS_TYPE
- TS_DSSIZE
- TS_HASHSPACE
- TS_NTABLES
- TS_ORGANIZATIONTYPE
- TS_PAGENUM
- TS_PARTITIONS
- TS_PGSIZE
- TS_SEGSIZE
- TS_TYPE

Note: If you specify IX_COLNO, IX_COLSEQ, and IX_ORDERING, you must also specify CHECK-INDEX-KEYS(YES).

returncoden must be 0, 4, or 8.

- Default: If the mismatch return code command is not specified, the default return code is 4.
- Required: No
- Restrictions: None
- Short form: OMMRC

OBJECT-TRANSLATE(

DATABASE, *sourcedatabase, targetdatabase*

[DATABASE, *sourcedatabasen, targetdatabasen*]

TABLESPACE, *sourcetablespace1, targettablespace1 . . .*

[TABLESPACE, *sourcetablespacen, targettablespacen*]

TABLE, *sourcetable1, targettable1 . . .*

[TABLE, *sourcetablen, targettablen*]

INDEXSPACE, *sourceindexspace1, targetindexspace1 . . .*

[INDEXSPACE, *sourceindexspacen, targetindexspacen*]

INDEX, *sourceindex1, targetindex1 . . .*

[INDEX, *sourceindexn, targetindexn*]

CREATOR, *sourcecreator1, targetcreator1 . . .*

[CREATOR, *sourcecreatorn, targetcreatorn*]

VCAT, *sourcevcat1, targetvcat1 . . .*

[VCAT, *sourcevcatn, targetvcatn*]

The OBJECT-TRANSLATE parameter allows Db2 Cloning Tool Table Space Cloning to map the source objects to target objects with different names. This feature allows Db2 Cloning Tool Table Space Cloning to copy table spaces and index spaces to the same subsystem or to a different subsystem with different names.

The DATABASE, INDEXSPACE, TABLESPACE, INDEX and TABLE parameters specify the names of the source objects and the new target object names, in pairs.

The CREATOR parameter specifies the creator name of the source objects (databases, table spaces, index spaces, tables, or indexes) and the new creator name for the corresponding target objects.

The VCAT parameter specifies the VCAT name of the source table spaces or index spaces and the new VCAT name for the target table spaces or index spaces. Note that DEFVCAT in the TARGET-DB2 parameter is still used if a VCAT is unavailable after applying all OBJECT-TRANSLATE VCAT commands. VCAT mapping is used only when the target objects do not exist and the target objects will not be created using DDL generation. If the target objects will be created using DDL generation and you want to change the VCAT for the target objects, DDL-ATTRIBUTE-CHANGE VCAT rules must be specified.

Db2 Cloning Tool Table Space Cloning will verify the source names on the source Db2. When found, the supplied target names are used to verify compatible target table spaces and index spaces on the target Db2.

If the INDEXSPACE parameter is specified, Db2 Cloning Tool Table Space Cloning uses the Database.Indexspace name to find the target index. If INDEXSPACE is not specified, then the Creator.Name is used to find the target index. The INDEXSPACE parameter is not recommended for OBJECT-TRANSLATE, because it is common for indexspace names for implicit indexes or for indexes with long names to be different for the source and the target. Db2 Cloning Tool Table Space Cloning can match implicit and explicit indexes with their associated tables without the need to specify the INDEXSPACE parameter.

The source and object pairs may be specified using masking. Specify masking as follows:

- Percent sign (%) or asterisk (*) represents n characters.
- Underscore (_) or question mark (?) represents a single character. Use the question mark (?) rather than the underscore (_) for creator, table and index names, as the underscore is a valid character for these three object names.

Refer to the topic [“OBJECT-TRANSLATE considerations”](#) on page 554 for additional information.

- Default: None
- Required: No
- Restrictions: None
- Short form: OBJXLATE

PROCESS-DDL(DDL-ENABLE(Y | N)
[,PROCESS-TYPE([Y | N | G | X | A])]
[,PROCESS-DDL-DDN(*ddname*)]
[,COMMIT-FREQUENCY(*numberofstatements*)]
[,DDL-ATTRIBUTE-CHANGE
(*attributename,srcvalue,trgvalue,applyTOTYPE,applyTOOBJECT*)]]
[,EXPLODE-OBJECTS(Y | N)]
[,GENERATE-DDL-DEFAULTS(Y | N)]
[,IGNORE-CREATE-OBJECT-EXISTS(Y | N)]
[,IGNORE-DROP-OBJECT-DOES-NOT-EXIST(Y | N)]
[,RECREATE-OBJECTS([A | M | N])]
[,RECREATE-TABLE-RESTRICTED-ON-DROP(Y | N)]
[,SUPPRESS-GRANTS(Y | N)]
[,SUPPRESS-RI-CONSTRAINTS(Y | N)]
[,USE-DDL-SQLID(*sqlid*)]

PROCESS-DDL generates and/or executes DDL to be used for creating missing or for re-creating existing target objects. DROP DDL can be generated for tables, table spaces and indexes. CREATE DDL can be generated for storage groups, databases, table spaces, tables, indexes, views, aliases, synonyms, triggers, stored procedures, functions, data types, sequences, sequence aliases, global variables, and materialized query tables. LOB and XML spaces are supported. All referenced storage groups, distinct types and other supporting objects must exist on the target to be able to execute the generated DDL. In addition to supporting missing target objects, Db2 Cloning Tool Table Space Cloning can generate source object DDL to be saved to a data set or can execute DDL from an input data set.

For more information about the DDL generation process, refer to the topic [“Considerations for generating target object DDL using PROCESS-DDL”](#) on page 195.

- Default: N
- Required: No
- Restrictions: None
- Short form: DDL

DDL-ENABLE - Set this parameter to Y to enable DDL processing. If set to N, DDL processing is disabled; this allows you to leave the DDL commands in the source JCL without processing them.

- Default: N
- Required: No
- Restrictions: None
- Short form: DDLENA

PROCESS-TYPE - Specify the processing type.

- Y: Generate and execute DDL for non-existent target objects.
- N: Do not generate or execute DDL. This is the default.
- G: Generate DDL for missing target objects, but do not execute.
- X: Do not generate DDL, but execute DDL already in PROCESS-DDL-DDN(*ddname*).
- A: Generate all source object DDL, but do not execute.

Generated DDL is always written to the DD name that is provided in the PROCESS-DDL-DDN(*ddname*) parameter. The PROCESS-DDL-DDN is required.

To prevent copies to the target subsystem using the F0001 data sets, run PROCESS-TYPE(G) with PGM(NONE). Once the target DDL is correct, then submit the copy jobs.

If the processing type is Y or G, and multiple VCATs are required to map data set names from source to target, you must specify the DDL-ATTRIBUTE-CHANGE VCAT parameter.

If you specify PROCESS-TYPE(A), Db2 Cloning Tool reads the source catalog and generates the source DDL. No connection to the target system is made and no copies are made. If you want Db2 Cloning Tool to read the target catalog and compare source and target objects, specify the PGM(NONE) or SIM(A) parameters without source DDL generation enabled.

- Default: N
- Required: No
- Restrictions: None
- Short form: PTYPE

PROCESS-DDL-DDN - Specify the DD name where the generated DDL will be written to and read from. The DD name CKZDDLMS cannot be specified for PROCESS-DDL-DDN (or any other Db2 Cloning Tool DD). If objects with identity columns are cloned, the PROCESS-DDL block should be specified with PROCESS-DDL-DDN; PROCESS-TYPE and DDL-ENABLE may be set to N.

- Default: None
- Required: Yes, unless PROCESS-TYPE is N and no objects with identity columns are cloned.
- Restrictions: None
- Short form: DDL-DDN

COMMIT-FREQUENCY – This parameter specifies the number of statements after which a COMMIT statement is generated. Regardless of the specified value, a COMMIT is generated after the DDL is generated for all of the objects of a specific object type. For example, a COMMIT is generated after the DDL is generated for all table spaces, but before the table DDL is generated.

- Default: 1000
- Required: No
- Restrictions: Valid values are 1-1000.
- Short form: COMF

DDL-ATTRIBUTE-CHANGE - This parameter allows values not changed with object translate to be changed from source to target. Refer to the topic [“DDL-ATTRIBUTE-CHANGE parameter values”](#) on page 555 for a detailed explanation of this parameter.

- Default: None
- Required: No
- Restrictions: None
- Short form: DDLAC

EXPLODE-OBJECTS - When set to Y, DDL generation reads the objects to be cloned, including their dependencies and requirements, from the catalog and generates CREATE statements for all objects that are not found on the target. When set to N, only DDL for the objects to be cloned is generated.

- Default: N
- Required: No
- Restrictions: None
- Short form: EXPLO

GENERATE-DDL-DEFAULTS - When set to Y, this parameter generates DDL statements for default values. If set to N, the statements are omitted. For example, if GENERATE-DDL-DEFAULTS is set to N and the catalog value for CLOSE is Y (the default), CLOSE YES is not included in the DDL stream. If GENERATE-DDL-DEFAULTS is set to Y, CLOSE YES is included in the DDL stream.

- Default: N
- Required: No

- Restrictions: None
- Short form: GDDL

IGNORE-CREATE-OBJECT-EXISTS - This parameter can be used when PROCESS-TYPE is X or Y. If IGNORE-CREATE-OBJECT-EXISTS is set to Y, -601 SQL errors are ignored.

- Default: N
- Required: No
- Restrictions: None
- Short form: IGOBJ

IGNORE-DROP-OBJECT-DOES-NOT-EXIST - This parameter can be used when PROCESS-TYPE is X or Y. If IGNORE-DROP-OBJECT-DOES-NOT-EXIST is set to Y, -204 SQL errors are ignored.

- Default: N
- Required: No
- Restrictions: None
- Short form: IDRONE

RECREATE-OBJECTS - Specify which objects are to be re-created as follows:

- A: All target objects and their dependent objects (such as indexes and views) that are paired to source objects are dropped and re-created using the source objects as a model.
- M: Only objects with mismatches are re-created.
- N: No objects are re-created.

If RECREATE-OBJECTS is set to A or M, parameters ALWAYS-COPY-INDEXSPACES and INCLUDE-ALL-RI are automatically set to Y.

- Default: N
- Required: No
- Restrictions: RECREATE-OBJECTS(A) and RECREATE-OBJECTS(M) are valid only for PROCESS-TYPE(G) and PROCESS-TYPE (Y).
- Short form: RECROBJ

RECREATE-TABLE-RESTRICTED-ON-DROP- This parameter controls behavior when re-creating tables with RESTRICT ON DROP. When this keyword is set to Y, ALTER statements are generated to remove DROP restrictions from re-created tables. When this keyword is set to N, tables with RESTRICT ON DROP are marked as not eligible for re-creation and are not re-created. This keyword is ignored when RECREATE-OBJECTS is N.

- Default: N
- Required: No
- Restrictions: None
- Short form: RTROD

SUPPRESS-GRANTS - This parameter allows you to choose whether the GRANT statements are generated for the target system during table space cloning. If SUPPRESS-GRANTS is set to Y, GRANT statements are not included in the generated DDL, so you do not need to manually revoke privileges from cloned objects.

- Default: N
- Required: No
- Restrictions: None
- Short form: SUP-GR

SUPPRESS-RI-CONSTRAINTS - When set to Y, suppresses RI relationships and constraints in target DDL. When set to N, RI relationships and constraints are generated in target DDL. Only specify Y if you

ensure that RI relationships and constraints will remain intact during the clone, or the target data is read-only (and therefore formal RI is not required).

- Default: N
- Required: No
- Restrictions: None
- Short form: SUPPRI

USE-DDL-SQLID – This parameter specifies the SQLID to be used in the CURRENT SQLID statement in DDL generation. The SQLID that is provided can be up to eight characters long.

- Default: None
- Required: No
- Restrictions: None
- Short form: SQLIDDDL

RESET-LOGRBA(Y | N | A)

The RESET-LOGRBA parameter is passed to the Db2 Cloning Tool Table Space Cloning target job and allows Db2 Cloning Tool Table Space Cloning to reset the LOGRBA. The LOGRBA will always be reset if there are OBID changes to be made. The level IDs in the target VSAM objects are always reset to prevent Db2 down-level rejection of the target VSAM objects. RESET-LOGRBA(A) performs an automatic reset of the LOGRBA depending on the conditions. When the source and target Db2 subsystems are in the same Db2 subsystem, and RESET-LOGRBA is set to A, the LOGRBA is not reset. When the source and target Db2 subsystems are in different Db2 subsystems, and RESET-LOGRBA is set to A, the LOGRBA is reset. RESET-LOGRBA(A) is required when cloning universal table spaces in order to optimize page processing.

- Default: Directed by CKZINI token RESET_LOGRBA in the PARMLIB :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: RL

RTS-COPY (RTS-COPY-ENABLE(Y | N)

[,DELETE-RTS-DATASETS (Y | N)]
[,RTS-DETAILS(Y | N)]
[,RTSFILE-DATA-SET-HLQ(*dsname*)]
[,RTSFILE-UNIT-TYPE(SYSALLDA)]
[,RTSFILE-QUANTITY-IN-TRACKS(Y | N)]
[,RTSFILE-PRIMARY-QUANTITY(*nnnn*)]
[,RTSFILE-SECONDARY-QUANTITY(*nnnn*)]
[,RTSFILE-DATACLAS(*dataclas*)]
[,RTSFILE-STORCLAS(*storclas*)]
[,RTSFILE-MGMTCLAS(*mgmtclas*)]

This parameter allows you to specify options for copying real-time and RUNSTATS statistics from source to target as part of the cloning process. Db2 Cloning Tool Table Space Cloning reads statistics from the Db2 catalog tables and writes them to temporary data sets in the source job. In the target job, Db2 Cloning Tool Table Space Cloning reads statistics from temporary data sets and writes them to the Db2 catalog tables.

- Default: N
- Required: No
- Restrictions: None
- Short form: RTSC

RTS-COPY-ENABLE - Set this parameter to Y to specify that real-time and RUNSTATS statistics of source objects are to be copied.

- Default: N
- Required: No
- Restrictions: None
- Short form: RTS-ENABLE

RTS-DETAILS - Set this parameter to Y to print a full real-time and RUNSTATS statistics report.

- Default: N
- Required: No
- Restrictions: None
- Short form: RTS-DTL

RTSFILE-DATA-SET-HLQ – Specify the high level qualifier for temporary data sets that will be allocated to copy the real-time and RUNSTATS statistics of source objects.

- Default: CKZ.RTSDATA
- Required: No
- Restrictions: None
- Short form: RTS-HLQ

RTSFILE-UNIT-TYPE - Specify the unit type for the real-time and RUNSTATS statistics data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: RTS-UT

RTSFILE-QUANTITY-IN-TRACKS - Specify Y if the real-time and RUNSTATS statistics data set is to be allocated in tracks or N if the data set is to be allocated in cylinders.

- Default: Y
- Required: No
- Restrictions: None
- Short form: RTS-TRK

RTSFILE-PRIMARY-QUANTITY - Specify the primary quantity for the real-time and RUNSTATS statistics data set.

- Default: 100
- Required: No
- Restrictions: None
- Short form: RTS-PQT

RTSFILE-SECONDARY-QUANTITY - Specify the secondary quantity for the real-time and RUNSTATS statistics data set.

- Default: 100
- Required: No
- Restrictions: None
- Short form: RTS-SQT

RTSFILE-DATACLAS - If the real-time and RUNSTATS statistics data set will be managed by SMS, specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None

- Short form: RTS-DATA

RTSFILE-STORCLAS - If the real-time and RUNSTATS statistics data set will be managed by SMS, specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: RTS-STOR

RTSFILE-MGMTCLAS - If the real-time and RUNSTATS statistics data set will be managed by SMS, specify the SMS Management Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: RTS-MGMT

DELETE-RTS-DATASETS – Specify Y if you want the temporary data set that is produced by real-time and RUNSTATS statistics processing to be deleted when processing is complete.

- Default: N
- Required: No
- Restrictions: None
- Short form: DEL-RTS

SIMULATE(A | N | Y)

The SIMULATE parameter allows Db2 Cloning Tool Table Space Cloning to be run in one of following modes.

SIM(N) is normal or non-simulation mode.

SIM(Y) performs basic simulation. Use SIM (Y) to check syntax on all the source job parameters in the source job and verify source objects. With SIM (Y), Db2 Cloning Tool Table Space Cloning confirms that all of the source objects that are specified in the LISTDEF exist, and lists the source objects in a report. No action on or reporting of target objects or target job syntax is done, and no copies are performed.

SIM(A) performs the same syntax checking and source object checking as SIM(Y), but also checks for mismatches between source and target objects and checks target objects and data sets. With SIM(A), Db2 Cloning Tool Table Space Cloning allocates the existing target data sets, and verifies the target catalog and physical data sets. If PGM(ADRDSSU) or PGM(EMCAPI), Db2 Cloning Tool Table Space Cloning stops target and then source spaces, calls the data mover program in NORUN mode, starts source and target spaces and writes out SYNCDB2 commands for the target. If PGM(NONE), Db2 Cloning Tool Table Space Cloning validates the target table spaces and index spaces and writes out SYNCDB2 commands for the target. No copies are performed. No target object pages are changed; the target job SCAN-ONLY parameter is set to Y when the job is run.

- Default: N
- Required: No
- Restrictions: SIM(A) is not valid with DATA-MASKING(Y), PGM(SRCIMCPY), PGM(SRCVSCPY), SUBTASK-DATASET-EXTENSIONS(Y), or USE-RUNTIME-REPOSITORY(Y).
- Short form: SIM. A blank value (SIM()) or an empty value (SIM()) are treated as SIM(Y).



CAUTION: If you do not want DDL to be processed during a simulation, change DDL-ENABLE to N, or change PROCESS-TYPE to G and write the DDL to a file.

START-SOURCE-DDN(ddname)

The START-SOURCE-DDN parameter specifies the name of a data set where Db2 Cloning Tool Table Space Cloning will write out Db2 START commands for all the source table spaces and index spaces. The data set pointed to by START-SOURCE-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in copying the VSAM objects outside of Db2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER PGM(NONE) is specified.
- Short form: STARS-DDN

STOP-SOURCE-DDN(ddname)

The STOP-SOURCE-DDN parameter specifies the name of a data set where Db2 Cloning Tool Table Space Cloning will write out Db2 STOP commands for all the source table spaces and index spaces. The data set pointed to by STOP-SOURCE-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in copying the VSAM objects outside of Db2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER PGM(NONE) is specified.
- Short form: STOPS-DDN

STOP-TARGET-DDN(ddname)

The STOP-TARGET-DDN parameter specifies the name of a data set where Db2 Cloning Tool Table Space Cloning will write out Db2 STOP commands for all the target table spaces and index spaces. The data set pointed to by STOP-TARGET-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in copying the VSAM objects outside of Db2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER PGM(NONE) is specified
- Short form: STOPT-DDN

SYNCDB2-DDN(ddname)

The SYNCDB2-DDN parameter supplies the ddname assumed via JCL to point at the data set which will be passed to the Db2 Cloning Tool Table Space Cloning target job for ID translation and to make the VSAM objects accessible to the target Db2. This parameter is required if the target job must be run to reset LOG RBAs or change object IDs or both. The data set pointed to by SYNCDB2-DDN must have an LRECL of 80 and RECFM of FB.

- Default: None
- Required: No. This is the input to the target job. Required only if any changes, such as data masking, new OBIDs or resetting the LOGRBA, are to be made to the target data sets after being copied.
- Restrictions: None
- Short form: SYN-DDN

TARGET-JOB-INDEX-REBUILD-DDN(ddname)

TARGET-JOB-INDEX-REBUILD-DDN can be used to rebuild all indexes whose tables were affected by page changes. Any table in the target job that has a page changed via data masking requires its indexes to be rebuilt. Also, if INDEX-LOG-APPLY is set to N, any table in the target job that has a page changed via log apply requires its indexes to be rebuilt. TARGET-JOB-INDEX-REBUILD-DDN specifies the input and output DDs to be used to generate REBUILD INDEX utility jobs.

To use this functionality, you must modify a sample template provided by Db2 Cloning Tool Table Space Cloning and add DDs to the target job. For detailed instructions on using this keyword to rebuild indexes, refer to the topics Chapter 17, [“Using data masking with table space cloning,”](#) on page 271, Chapter 19, [“Using LOG-APPLY to make consistent copies of table spaces and index spaces,”](#) on page 293, and Chapter 20, [“Options for rebuilding indexes,”](#) on page 297. When the target job completes, submit the output to rebuild all affected indexes.

- Default: None
- Required: No
- Restrictions: None
- Short form: TRGIR

TARGET-JOB-REPAIR-DDN(*ddname*)

TARGET-JOB-REPAIR-DDN is used to generate a REPAIR job to correct differences between the Db2 catalog and one or more page sets, and to specify whether Db2 Cloning Tool Table Space Cloning or the user submits the job. Specify one to seven characters of the target job DD name; the DDNAME is suffixed with .I and .O when added to the template. The default *ddname* value is CKZTRP, as provided in the Tools Customizer step that allocates table space cloning data sets (see [“Task: Application Cloning \(Table Space Cloning\) tasks”](#) on page 53).

- Default: None
- Required: No
- Restrictions: None
- Short form: TRGJR

UNLOAD-LOAD (UNLOAD-LOAD-ENABLE(Y | N)

[,DELETE-DATASETS(Y | N)]

[,LOAD-REUSE(Y | N)]

[,LOAD-SORTNUM(*integer*)]

[,UNLOAD-LOAD-EARLY-VALIDATION(Y | N)]

,TEMPLATE-SORTOUT-DDN(*ddname*)

,TEMPLATE-SYSPUNCH-DDN(*ddname*)

,TEMPLATE-SYSREC-DDN(*ddname*)

,TEMPLATE-SYSUT-DDN(*ddname*))

UNLOAD-LOAD is used in the source job and allows unload and load processing to occur when any of the following source and target table space attributes do not match:

- TS_CLONE
- TS_COMPRESS
- TS_COMPRESS_TYPE
- TS_DSSIZE
- TS_PAGENUM
- TS_PARTITIONS
- TS_PGFSIZE
- TS_SEGSIZE
- TP_COMPRESS
- TP_DSSIZE
- TP_LIMITKEY
- CL_LENGTH (only if the length of the source column is less than the length of the target column)

The remainder of the possible mismatches are considered incompatible with UNLOAD/LOAD.

UNLOAD-LOAD processing invokes the DSNUTILU or DSNUTILV stored procedure that is supplied with Db2. Therefore, a WLM environment for DSNUTILU or DSNUTILV must be configured and available. Refer to IBM Knowledge Center for your version of Db2 for more information about DSNUTILU or

DSNUTILV. For information about WLM environment requirements, see IBM Knowledge Center topic *Core WLM environments for Db2-supplied routines*.

When UNLOAD-LOAD-ENABLE is set to Y and UNLOAD-LOAD-EARLY-VALIDATION is set to Y, before the cloning the source job verifies that SORTOUT, SYSPUNCH, SYSREC, and SYSUT templates with control statements for UNLOAD-LOAD processing are available and accessible. In addition, the source job completes basic validation of UNLOAD-LOAD templates by calling the UNLOAD-LOAD utilities in preview mode before the cloning.

UNLOAD-LOAD processing automatically detects if the DSNUTILV stored procedure is available on the source and target Db2 subsystems. If DSNUTILV is available on a Db2 subsystem, Db2 Cloning Tool will use it to perform UNLOAD/LOAD. If DSNUTILV is not available, then Db2 Cloning Tool will use DSNUTILU. When DSNUTILV is available on one of Db2 subsystem, then DSNUTILV will be used where it is available, and DSNUTILU will be used where DSNUTILV is not available.

UNLOAD-LOAD processing takes place when mismatches are encountered as follows:

- If EXCLUDE-MISMATCH-PROCESSING(Y) is specified, then all objects are processed as if there are no mismatches, regardless of the setting for UNLOAD-LOAD-ENABLE.
 - If UNLOAD-LOAD-ENABLE(Y) and OBJECT-MISMATCH-RETURN-CODE is 0, then an informational message is issued and the object is treated as if there is no mismatch, regardless of the setting for UNLOAD-LOAD-ENABLE.
 - If UNLOAD-LOAD-ENABLE(Y), OBJECT-MISMATCH-RETURN-CODE is 4, and the mismatch is a type that can be processed by UNLOAD-LOAD, then UNLOAD-LOAD is used, regardless of the value of ALLOW-COPY-ON-MISMATCH. See the topic [“How Db2 Cloning Tool Table Space Cloning checks object compatibility”](#) on page 192 for additional information about this case.
 - If UNLOAD-LOAD-ENABLE(Y), OBJECT-MISMATCH-RETURN-CODE is 4, and the mismatch cannot be processed by UNLOAD-LOAD, then a warning message is issued.
 - If UNLOAD-LOAD-ENABLE(Y) and OBJECT-MISMATCH-RETURN-CODE is 8, an error message is issued and no copies are performed, regardless the setting for UNLOAD-LOAD-ENABLE.
- Default: None
 - Required: No
 - Restrictions:
 - The maximum size of the DSNUTILU utility control statement buffer is 32704 bytes and the maximum allowed size of the DSNUTILV utility control statement buffer is 1 MB. This buffer is used by table space cloning to pass unload and load utility control statements to the DSNUTILU and DSNUTILV stored procedures. An error message is issued when the capacity of the utility control statement buffer is exceeded.
 - SYSREC data set spanned record format (VBS) is not supported.
 - Objects with referential integrity relationships are not supported.
 - UNLOAD-LOAD functionality is not supported for clone instances of objects if their base objects are to be processed by UNLOAD-LOAD. In that case, you should exclude the clone instances from the cloning job.
 - Short form: UL

UNLOAD-LOAD-ENABLE - Set this parameter to Y to specify that objects are to be unloaded and loaded when an object attribute mismatch is detected.

- Default: N
- Required: No
- Restrictions: None
- Short form: UL-ENABLE

UNLOAD-LOAD-EARLY-VALIDATION - Set this parameter to Y to specify that if UNLOAD-LOAD-ENABLE is Y, before the cloning the source job verifies that SORTOUT, SYSPUNCH, SYSREC, and SYSUT templates with control statements for UNLOAD-LOAD processing are available and accessible.

In addition, the source job completes basic validation of UNLOAD-LOAD templates by calling the UNLOAD-LOAD utilities in preview mode before the cloning.

- Default: N
- Required: No
- Restrictions: None
- Short form: UL-EVD

DELETE-DATASETS - Set this parameter to Y to specify that data sets that are produced by unload and load are to be deleted when an unload and load for an object is complete.

- Default: Y
- Required: No
- Restrictions: None
- Short form: UL-DEL

LOAD-REUSE - Set this parameter to Y to generate the REUSE option of the LOAD control statement.

- Default: Y
- Required: No
- Restrictions: None
- Short form: UL-REU

LOAD-SORTNUM - Specify the number of temporary data sets that are to be dynamically allocated by the sort program during LOAD processing. SORTNUM is ignored if UNLOAD does not generate the LOAD command option SORTDEVT. If specified, the value must be within the range of 2-255.

- Default: 2
- Required: No
- Restrictions: None
- Short form: UL-SNUM

TEMPLATE-SORTOUT-DDN - Specify the DD name of the file that contains the SORTOUT TEMPLATE control statement to be used during utility processing. This file must be defined with RECFM=FB and LRECL=80.

The TEMPLATE-SORTOUT-DDN must reference the file that contains the SORTOUT TEMPLATE control statement. Initially, the file with the SORTOUT TEMPLATE control statement can be found in the CKZULOUT member of the SCKZJCL sample library. Before cloning, copy the CKZULOUT member from the sample library to the TEMPLATE-SORTOUT-DDN member. You can modify the member as required.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: UL-TMP-SORTOUT

TEMPLATE-SYSPUNCH-DDN - Specify the DD name of the file that contains the SYSPUNCH TEMPLATE control statement to be used during utility processing. This file must be defined with RECFM=FB and LRECL=80.

The TEMPLATE-SYSPUNCH-DDN must reference the file that contains the SYSPUNCH TEMPLATE control statement. Initially, the file with the SYSPUNCH TEMPLATE control statement can be found in the CKZULPUN member of the SCKZJCL sample library. Before cloning, copy the CKZULPUN member from the sample library to the TEMPLATE-SYSPUNCH-DDN member. You can modify the member as required.

- Default: None
- Required: Yes

- Restrictions: None
- Short form: UL-TMP-SYSPUNCH

TEMPLATE-SYSREC-DDN - Specify the DD name of the file that contains the SYSREC TEMPLATE control statement to be used during utility processing. This file must be defined with RECFM=FB and LRECL=80.

The TEMPLATE-SYSREC-DDN must reference the file that contains the SYSREC TEMPLATE control statement. Initially, the file with the SYSREC TEMPLATE control statement can be found in the CKZULREC member of the SCKZJCL sample library. Before cloning, copy the CKZULREC member from the sample library to the TEMPLATE-SYSREC-DDN member. You can modify the member as required.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: UL-TMP-SYSREC

TEMPLATE-SYSUT-DDN - Specify the DD name of the file that contains the SYSUT TEMPLATE control statement to be used during utility processing. This file must be defined with RECFM=FB and LRECL=80.

The TEMPLATE-SYSUT-DDN must reference the file that contains the SYSUT TEMPLATE control statement. Initially, the file with the SYSUT TEMPLATE control statement can be found in the CKZULUT member of the SCKZJCL sample library. Before cloning, copy the CKZULUT member from the sample library to the TEMPLATE-SYSUT-DDN member. You can modify the member as required.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: UL-TMP-SYSUT

WARN-IF-OBJECT-NOT-TRANSLATED (Y | N)

The WARN-IF-OBJECT-NOT-TRANSLATED parameter specifies that Db2 Cloning Tool Table Space Cloning is to check that each source object is translated to a new target value. Source objects that are not translated result in a warning message. Use this with PGM(NONE) to check that all masks are set correctly.

- Default: N
- Required: No
- Restrictions: None
- Short form: WIOXT

WARN-ON-DATASET-EXTENSION-MISMATCH(Y | N)

The WARN-ON-DATASET-EXTENSION-MISMATCH parameter provides the ability to specify the type of message that is issued when the number of data set extensions differs from source to target. If you specify WARN-ON-DATASET-EXTENSION-MISMATCH(Y), a warning message is issued and the job return code is set to 4. If you specify N, an informational message is issued.

- Default: N
- Required: No
- Restrictions: None
- Short form: WODXM

WARN-ON-INCOMPLETE-RI (Y | N)

The WARN-ON-INCOMPLETE-RI parameter specifies that when one or more LISTDEF statements do not specify RI, a warning message is issued.

For more information about how this keyword affects LISTDEF processing, see the topic [“Impact of COPY command keywords on LISTDEF processing”](#) on page 562.

- Default: N
- Required: No
- Restrictions: None
- Short form: WARNONICRI

WARN-ON-SIMPLE-TABLESPACE (Y | N)

This parameter allows you to be warned when simple table spaces are to be copied. When N is specified, an informational message is printed for each simple table space found on the source or target subsystem. When Y is specified, a warning message is issued.

In some cases when a simple table space is copied by Db2 Cloning Tool Table Space Cloning, the copy may have duplicate rows. This is caused by incompatibilities between table spaces migrated from an earlier Db2 version and processed using Db2 Version 9.1 or later.

When a simple table space is copied for the first time, ensure the target table space is accessible and has the correct number of rows. Once all of the simple table spaces in a job are verified, use the default N to eliminate warning messages.

- Default: N
- Required: No
- Restrictions: None
- Short form: WOSTS

OBJECT-TRANSLATE considerations

This topic describes specifics about using the OBJECT-TRANSLATE keyword.

Mask syntax

The source and object pairs in the OBJECT-TRANSLATE keyword may specify masking. The table that follows shows the allowable filter characters.

| <i>Table 87. Filter characters allowed for source and object pairs filter masks</i> | |
|---|---|
| Character | Description |
| * or % | A single asterisk or percent sign represents 0 to nn characters of any value. |
| _ or ? | An underscore or question mark represents one non-blank character. Use the question mark (?) rather than the underscore (_) for creator, table, and index names, as the underscore is a valid character for these object names. |

When using masks, the order of the object translate specifications for each object type becomes critical. Masking must be in the order of most restrictive to least restrictive. If not, the wrong target name could be selected for the corresponding source name.

Other masking rules are:

- The order, number, and type of mask characters must be the same between the source and target object
- %%%, %_, or _% are invalid masks if neither the source nor target has a non-masking character in between. However, a double underscore (__) is a valid mask.

The following table shows examples of valid and invalid masks for the OBJECT-TRANSLATE keyword.

| <i>Table 88. Masks for the OBJECT-TRANSLATE keyword</i> | | | |
|---|--------------------|-------------------------|--|
| Source Mask | Target Mask | Valid or Invalid | Comments |
| %aa_a% | _aaa_a% | Invalid | Source begins with %, target begins with _ |
| _aa_a% | _aaa_a | Invalid | Source ends with %, target does not end with % |

Table 88. Masks for the OBJECT-TRANSLATE keyword (continued)

| Source Mask | Target Mask | Valid or Invalid | Comments |
|-------------|-------------|------------------|---|
| _aa_% | _aaa_% | Invalid | Neither source nor target has a character between _ and % |
| _aa%%% | _aaa%%% | Invalid | Neither source nor target has a character between %% |
| _aa_% | _aaa_a% | Valid | |
| _aa_a% | _aaa_% | Valid | |
| _aa%_ | _aaa%a_ | Valid | |
| _aa%a% | _aaa%%% | Valid | |
| _aa_% | _aaa_a% | Valid | |
| aa___a | aa_a__a | Valid | |
| aa_a__a | aa___a | Valid | |

Syntax example

```
OBJECT-TRANSLATE(
CREATOR,%00?_,%0N?_,
TABLESPACE,Z@50_T___,Z#50_U___,
DATABASE,Z@50_DB,Z@50_EB,
DATABASE,Z@50_DB_,Z@50_DB_,
DATABASE,Z#50%B_,Z#50%C_,
DATABASE,Z@50_D%,Z#50_D%,
CREATOR,%N?_,%N?_,
TABLE,%TB11,%TB12,
INDEX,Z%,Y%
VCAT,%P,%T
)
```

Specifying a database object translate mask and including indexspaces in LISTDEF

In some cases, when using object translation to change a database name, target indexspaces may not be found. This is because Db2-generated indexspace names may not necessarily be identical from the source to the target subsystems. For example, INDEXA on the source may not have the identical indexspace name as INDEXA on the target. The translated database cannot be used unless an indexspace object translation value for the indexspace is provided with the translated database. With masking available for object translate, if all the indexspace names are the same from the source to the target, specify:

```
OBJECT-TRANSLATE (database,a,b,indexspace,%,%)
```

and *database.indexspace* will be used to locate all the target indexspaces. If all the indexspace names are not the same across subsystems, then specify the indexspaces that are different before %, % in the object translate command; for example:

```
OBJECT-TRANSLATE
(database,a,b,indexspace,abc,efg,indexspace,jkl,mno,indexspace,%,%)
```

DDL-ATTRIBUTE-CHANGE parameter values

The DDL-ATTRIBUTE-CHANGE parameter allows values not changed with object translate to be changed from source to target. While OBJECT-TRANSLATE commands allow the pairing of source to target objects

for copying purposes, DDL-ATTRIBUTE-CHANGE parameters allow other attributes of the selected objects to be changed.

The syntax for the DDL-ATTRIBUTE CHANGE parameter is:

```
DDL-ATTRIBUTE-CHANGE (attributename,srcvalue,trgvalue,applytotype,applytoobject)
```

The possible parameter values are detailed in the following section.

attributename

This parameter value names the attribute that can be changed. Refer to [Table 89 on page 557](#) for a list of possible values. *attributename* cannot be a mask.

srcvalue

This parameter identifies the source value to be changes. It can be blank; if left blank, the change applies to all source values of that attribute. *srcvalue* can be a mask only for STOGROUP, BUFFERPOOL, VCAT, and KEYLABEL attributes. For masking, % represents zero or more characters and ? represents one character. No other masking characters are allowed.

trgvalue

This parameter identifies the value that the attribute will be in the target. It can be blank; if left blank, the source values are changed to the default for that attribute, if one exists. *trgvalue* can be a mask only for STOGROUP, BUFFERPOOL, VCAT, and KEYLABEL attributes. For masking, % represents zero or more characters and ? represents one character. No other masking characters are allowed.

Note: There is no default for BUFFERPOOL or VCAT.

applytotype

This parameter specifies the type of object that the attribute change will be applied to. *applytotype* can be:

- DATABASE
- TABLESPACE
- TABLEPART
- TABLE
- INDEX
- INDEXPART
- STOGROUP

If not specified, *applytotype* defaults to all object types that the *attributename* applies to. *applytotype* cannot be a mask.

applytoobject

This parameter names the object(s) that this change applies to. *applytoobject* references the source object names. It defaults to % or all objects of the specified type. *applytoobject* can also be a mask. To specify all object names, a blank or % may be used. When using a mask, it must refer to one of the following objects: database, table space, table, or index. For masking, % represents zero or more characters and ? represents one character. When using TABLEPART or INDEXPART as the *applytotype*, use the TABLESPACE or INDEXSPACE name for *applytoobject*.

Note: Leaving both *applytotype* and *applytoobject* blank applies the change to all object types and names that the *attributetype* applies to. Therefore, instead of an underscore, a question mark must be used to represent a single mask character.

Note: If the specified rule contains an object type and attribute that cannot be changed for that object type in the current version of Db2, no error or warning is issued. The rule is not passed by the DDL generator, and changes are not applied.

The following table lists the *attributename* parameters, their possible values (*srcvalue* and *trgvalue*), and the type of object that the parameter may apply to (*applytotype*):

| <i>Table 89. Parameter values by attributename</i> | | |
|--|---|---|
| attributename | srcvalue/trgvalue | applytotype |
| STOGROUP | | Database, table space (Db2 V12 only), table space partition, index (Db2 V12 only), index partition, storage group |
| GBPCACHE | CHANGED ALL SYSTEM NONE | Table space (Db2 V12 only), table space partition The SYSTEM value is supported only for LOB table spaces. |
| | CHANGED ALL NONE | Index (Db2 V12 only), index partition |
| LOG | YES NO | Table space |
| PRIQTY | 1-7 decimal digits or -1 | Table space (Db2 V12 only), table space partition, index (Db2 V12 only), index partition |
| SECQTY | 1-7 decimal digits or -1 | Table space (Db2 V12 only), table space partition, index (Db2 V12 only), index partition |
| | Note: PRIQTY and SECQTY are matched using the catalog value of PQTY and SECQTYI, not what is entered using the DDL. DDL generation multiplies this catalog value by 4 to get the PRIQTY and SECQTY DDL values. For user managed data sets, RUNSTATS must have been run to populate the fields in the catalog. | |
| TRACKMOD | YES NO | Table space (Db2 V12 only), table space partition |
| CLOSE | YES NO | Table space, index |
| DATA_CAPTURE | NONE CHANGES | Table |
| BUFFERPOOL | <i>pool name</i> | Database, table space, index |
| VCAT | <i>vcatname</i> | Table space (Db2 V12 only), table space partition, index (Db2 V12 only), index partition, storage group |
| KEYLABEL | key label | Table, storage group |

The following table provides a cross reference of object type (*applytotype*) to the attribute that can be changed (*attributename*).

| <i>Table 90. Possible attributename parameters by object type</i> | |
|---|--|
| applytotype | attributename |
| Database | STOGROUP, BUFFERPOOL |
| Table space | STOGROUP (Db2 V12 only), GBPCACHE (Db2 V12 only), PRIQTY (Db2 V12 only), SECQTY (Db2 V12 only), TRACKMOD (Db2 V12 only), LOG, CLOSE, BUFFERPOOL, VCAT (Db2 V12 only) |
| Table space partition | STOGROUP, GBPCACHE, PRIQTY, SECQTY, TRACKMOD, VCAT |

Table 90. Possible attributename parameters by object type (continued)

| applytotype | attributename |
|-----------------|--|
| Table | DATA_CAPTURE, KEYLABEL |
| Index | STOGROUP (Db2 V12 only), GBPCACHE (Db2 V12 only), PRIQTY (Db2 V12 only), SECQTY (Db2 V12 only), CLOSE, BUFFERPOOL, VCAT (Db2 V12 only) |
| Index partition | STOGROUP, GBPCACHE, PRIQTY, SECQTY, VCAT |
| Storage group | STOGROUP, VCAT, KEYLABEL |

Additional information about specifying DDL-ATTRIBUTE-CHANGE

DDL-ATTRIBUTE-CHANGE commands should be in order from the most restrictive to the least restrictive. Restrictions are specified using the *applytoobject* or mask. For example, assume the following two commands where all table space objects beginning with ZYX are included in the *applyto* mask, and that they be matched with the first LOG command:

```
DDL-ATTRIBUTE-CHANGE (          -
LOG, , YES, TABLESPACE, ZYX%,  -
LOG, , NO, , ,                  -
)
```

The table spaces will all be changed to LOG YES or remain LOG YES. Therefore, the second LOG command will change all those table spaces not beginning with ZYX from LOG YES to LOG NO, or they will remain LOG NO. Note that in this example, including "TABLESPACE" as the *applytotype* is not required because LOG can only be specified for table spaces. Most other attributes pertain to multiple object types and should be explicitly specified. For example, to only change GBPCACHE for index spaces, the INDEXSPACE *applytotype* should be specified in the command.

PRIQTY and SECQTY are matched using the catalog value of PQTY and SECQTYI, not what is entered using DDL. DDL generation multiplies this catalog value by 4 to get the PRIQTY and SECQTY DDL values. For user managed data sets, RUNSTATS must have been run to populate the fields in the catalog.

Specification exceptions

- An individual partition cannot be referenced. All partitions are checked for any attribute changes. If, however only one partition has a particular GBPCACHE value, for example, and there is an applicable DDL-ATTRIBUTE-CHANGE GBPCACHE command, no other partition will be affected.
- If a table space has the LOG NO attribute, DATA_CAPTURE CHANGES cannot be specified.
- Space attributes cannot be changed for LOB or XML spaces.

If attribute changes are required that are not supported by Db2 Cloning Tool Table Space Cloning, in the source job specify DDL(G) to generate the target DDL and PGM(NONE). After this source job has run, edit the DDL and then submit on the target subsystem. Then rerun the source job to access the new target objects.

GLOBAL command values for EMC TimeFinder/Clone Mainframe Snap Facility data set level support

This topic discusses considerations for EMC's GLOBAL command values when using the TimeFinder/Clone Mainframe Snap Facility's data set level support to make the copies.

GLOBAL command parameters

When Db2 Cloning Tool Table Space Cloning initiates EMC copies using the EMC API, most GLOBAL command parameter values use the EMC default. Exceptions are noted in the table that follows.

Table 91. GLOBAL parameter defaults setting by Db2 Cloning Tool Table Space Cloning

| GLOBAL parameter | Db2 Cloning Tool Table Space Cloning setting |
|---------------------------|--|
| NQWAIT | NO |
| NOTIFYWHENCOMPLETE | DATASET |
| REUSE | YES |
| TOLERATEALLOCATIONFAILURE | YES |
| TOLERATECOPYFAILURE | YES |
| TOLERATEREUSEFAILURE | YES |
| VERIFY | NO |
| VSAMENQMODE | NONE |

The GLOBAL values listed in the following table are set during processing using the Db2 Cloning Tool Table Space Cloning parameters.

Table 92. GLOBAL values set during processing by Db2 Cloning Tool Table Space Cloning

| EMC GLOBAL value | Db2 Cloning Tool Table Space Cloning setting |
|--------------------|--|
| HOSTCOPYMODE | If FUZZY-COPY(N), this parameter value is set to EXCLUSIVE. Otherwise, it is set to SHARED |
| MAXRC | This parameter always uses the Db2 Cloning Tool Table Space Cloning MAX-COPY-RC value. |
| REPLACE | This parameter is set to YES if REPLACE-TARGET-DSN(Y). Otherwise, it is set to NO. |
| TOLERATEENQFAILURE | If FUZZY-COPY(N), this parameter is set to NO. Otherwise, it is set to YES. |
| TYRUN | If SIM(A), this value is set to NORUN. Otherwise, it is set to RUN. |

If EMC GLOBAL values other than those previously described are required, then you may use JOB-TEMPLATES to create an EMC TimeFinder/Clone SNAP DATASET job and submit it after the source job has run. Unlike ADRDSSU, an EMC job template cannot be automatically submitted using the CMDDNAME parameter of the COPY DATA-MOVER command. That is due to the differences between the EMCAPI and EMC TimeFinder/Clone data set snap support. Refer to the JCL sample CKZJOB3 or CKZJOB8 (if the source objects are not stopped) for a sample of an EMC data set snap input template.

CKZJOB3 is similar to the EMCAPI job that is internally submitted by Db2 Cloning Tool Table Space Cloning. This template assumes the following Db2 Cloning Tool Table Space Cloning COPY parameters are in effect:

```
FUZZY-COPY(N)
MAX-COPY-RC(0)
```

CKZJOB8 is similar to the EMCAPI job that is internally submitted by Db2 Cloning Tool Table Space Cloning when FUZZY-COPY(Y) is specified. This template assumes the following Db2 Cloning Tool Table Space Cloning COPY parameters are in effect:

```
FUZZY-COPY(Y)
MAX-COPY-RC(0)
```

Other changes to the Db2 Cloning Tool Table Space Cloning COPY command that must be made are:

- Set COPY subcommand DSNS-PER-COPY to 1.
- Set COPY subcommand DSS-COPY-COMMANDS to the number of data sets you wish to copy in a single SNAP job step. The maximum is 256, which is the recommended setting.

Change your input template variable and submit the source job to create the output job template. After the Db2 Cloning Tool Table Space Cloning source job has successfully run and you have verified that the SNAP parameters are correct, submit the output template job to perform the copies using EMC SNAP DATASET.

DATASUBTYPE

The DATASUBTYPE command is optional.

DATASUBTYPE may be used to specify the data subtype value (BIT, MIXED, or SBCS) needed by data masking.

The DATASUBTYPE command is useful when you are specifying data masking for a column that has been migrated from an earlier version of Db2 and has a blank for FOREIGNKEY in SYSIBM.SYSCOLUMNS.

If a source column has data masking specified and either the source or target has a blank for the FOREIGNKEY column in SYSCOLUMNS, the non-blank value will be used by data masking. This will not be reported as an object mismatch.

If both source and target have different non-blank values, a mismatch will be reported unless the source value is overridden by a DATASUBTYPE command. The source value for FOREIGNKEY will be used to perform data masking.

If the DATASUBTYPE command is found for a column, it overrides whatever FOREIGNKEY values exist in the catalog. If the source and target catalogs have the same non-blank value, but it differs from the DATASUBTYPE command value, the DATASUBTYPE command value still is used. No warning message will be issued.

If no FOREIGNKEY value is specified in the catalog or by command, SBCS will be used.

If the column specified in the masking definition is not a string column, any DATASUBTYPE command for this table and column is ignored. String columns are defined with TYPEDEF values CHAR, VARCHAR, LONGVAR, CLOB, DBCLOB and DISTINCT where the DATATYPEID is a string type.

DATASUBTYPE command syntax

DATASUBTYPE

Required keywords:

```
TABLECREATOR( tblcreator )
TABLENAME( tblname )
COLUMNNAME( colname )
SUBTYPE( B | M | S )
```

DATASUBTYPE command & keyword definitions

Required and conditionally required keywords are described first, followed by optional keywords

DATASUBTYPE

The DATASUBTYPE command is optional. This command may be used to specify the data subtype value. This value may be needed by data masking when a column has been migrated from an earlier version of Db2 and has a blank for FOREIGNKEY in SYSIBM.SYSCOLUMNS. If the DATASUBTYPE command is found for a column, it overrides whatever FOREIGNKEY values exist in the catalog.

- Default: None
- Required: No

- Restrictions: None

Required DATASUBTYPE keywords

TABLECREATOR (*creator-id*)

TABLENAME (*table-name*)

COLUMNNAME (*column-name*)

TABLECREATOR, TABLENAME, and COLUMNNAME identify the source column that will be masked.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: TBCR, TBNM, CLNM

SUBTYPE(B | M | S)

Indicates the data subtype to be assumed as the FOREIGNKEY in SYSCOLUMNS. Valid subtypes are B for BIT, M for MIXED, or S for SBCS.

- Default: S
- Required: No
- Restrictions: None
- Short form: SUBT

HLQDDDF

The HLQDDDF command is optional.

HLQDDDF allows Db2 Cloning Tool Table Space Cloning to pass input and output DDs to ADRDSSU.

HLQDDDF command syntax

HLQDDDF

Required keywords:

```
DDNAME( dd1,...ddn )
HLQNAME( hlq1 )
optional keywords . . .
```

Optional keywords:

```
[ DIRECTION( IN | OUT ) ]
```

HLQDDDF command and keyword definitions

Required and conditionally required keywords are described first, followed by optional keywords

HLQDDDF

The HLQDDDF command is optional. HLQDDDF passes input and output DDs to ADRDSSU.

- Default: None
- Required: Yes
- Restrictions: None

Required HLQDDDF keywords

DDNAME (*vdd1,...ddn*)

DDNAME provides the DD specification to pass to ADRDSSU as an input or output DD.

- Default: None
- Required: Yes
- Restrictions: None

HLQNAME(*hlq*)

The HLQNAME parameter specifies the high level qualifier this command applies to.

- Default: None
- Required: Yes
- Restrictions: None

Optional HLQDDDF keywords

DIRECTION(IN | OUT)

DIRECTION specifies whether this DD name or names is provided to ADRDSSU as an input or output DD.

- Default: OUT
- Required: No
- Restrictions: None
- Short form: DIR

LISTDEF

The LISTDEF command is required. LISTDEF selects all Db2 table spaces and index spaces to be involved in the cloning process.

LISTDEF processing for Db2 Cloning Tool Table Space Cloning differs from IBM's LISTDEF in the following ways:

1. LIST is not supported.
2. You cannot change the name of the LISTDEF DD. It is CKZLSTDF. That DD must point to a single member with the LISTDEF control statements. The DD that is supplied must be LRECL=80 and RECFM=FB.
3. A detailed explanation of the LISTDEF processing is available when CKZLOG DD is specified for output. This includes output from the IBM LISTDEF utility.
4. LISTDEF output is available in the CKZPRINT DD. It consists of the data sets and the tables referenced.
5. '/'* ' and '*/' can be used to enter comments. Note that these comment start and stop commands must be bounded by spaces.
6. The ARCHIVE option is not supported.

Masking rules are the same as for IBM LISTDEF.

Impact of COPY command keywords on LISTDEF processing

The following COPY command keywords affect LISTDEF processing:

- ALWAYS-COPY-HISTORY-TABLES
- ALWAYS-COPY-INDEXSPACES
- INCLUDE-ALL-RI
- WARN-ON-INCOMPLETE-RI

The order in which LISTDEF statements are constructed within the rules of the LISTDEF command keywords and the COPY command keywords is as follows:

1. **Resolve STOGROUP statements, if specified.** Each statement with a STOGROUP object specification is replaced with statements with the DATABASE object specification for each database in the specified storage group.
2. **Process the INCLUDE-ALL-RI keyword, if specified.** An RI indicator is inserted in every statement of the specified LISTDEF.
3. **Process the WARN-ON-INCOMPLETE-RI keyword, if specified.** If any statement in the LISTDEF does not have a specified RI indicator, a warning message is issued. If MAX-RC=4, processing continues. If MAX-RC=0, LISTDEF processing immediately ends with an error.
4. **Process the ALWAYS-COPY-HISTORY-TABLES keyword, if specified.** Each statement that does not have a HISTORY indicator is duplicated with the HISTORY indicator. Each statement that has a HISTORY indicator is duplicated immediately after without the HISTORY indicator. For example, if the following LISTDEF statements are provided:

```
INCLUDE TABLESPACES DATABASE DBNAME1 HISTORY
INCLUDE TABLESPACES TABLESPACE DBNAME.TSNAME
INCLUDE INDEXSPACES TABLESPACE DBNAME.TSNAME
```

After processing the ALWAYS-COPY-HISTORY-TABLES keyword, the statements are as follows:

```
INCLUDE TABLESPACES DATABASE DBNAME1 ALL HISTORY
INCLUDE TABLESPACES DATABASE DBNAME1 ALL
INCLUDE TABLESPACES TABLESPACE DBNAME.TSNAME ALL
INCLUDE TABLESPACES TABLESPACE DBNAME.TSNAME ALL HISTORY
INCLUDE INDEXSPACES TABLESPACE DBNAME.TSNAME ALL
INCLUDE INDEXSPACES TABLESPACE DBNAME.TSNAME ALL HISTORY
```

5. **Process the ALWAYS-COPY-INDEXSPACES keyword, if specified.** All INCLUDE INDEXSPACES statements are deleted. Each statement with TABLESPACES type specification are duplicated right after with type specification INDEXSPACES. For example, if the following LISTDEF statements are provided:

```
INCLUDE TABLESPACES TABLESPACE DBNAME.TSNAME
INCLUDE INDEXSPACES INDEX CREATOR.INDEXNAME
INCLUDE TABLESPACES DATABASE DBNAME2
EXCLUDE INDEXSPACES INDEX CREATOR1.IXNAME1
```

First, INCLUDE INDEXSPACES rules are deleted, as follows:

```
INCLUDE TABLESPACES TABLESPACE DBNAME.TSNAME ALL
INCLUDE TABLESPACES DATABASE DBNAME2 ALL
EXCLUDE INDEXSPACES INDEX CREATOR1.IXNAME1
```

Then, all rules with TABLESPACES type specification are duplicated with INDEXSPACES type specification immediately after:

```
INCLUDE TABLESPACES TABLESPACE DBNAME.TSNAME ALL
INCLUDE INDEXSPACES TABLESPACE DBNAME.TSNAME ALL
INCLUDE TABLESPACES DATABASE DBNAME2 ALL
INCLUDE INDEXSPACES DATABASE DBNAME2 ALL
EXCLUDE INDEXSPACES INDEX CREATOR1.IXNAME1
```

Note: ALL is the default LOB specification for all statements, except for EXCLUDE INDEXSPACES INDEXSPACE and EXCLUDE INDEXSPACES INDEX; those statements do not use the ALL option unless you specify it on the LISTDEF statement. This allows the exclusion of only one index. If all indexes need to be excluded, specify the ALL parameter on the EXCLUDE INDEXSPACES INDEX/INDEXSPACE statements. If the EXCLUDE INDEXSPACES INDEXSPACE/INDEX statement is the result of processing the ALWAYS-COPY-INDEXSPACES keyword, its LOB specification is inherited from the parent statement. Therefore, if EXCLUDE TABLESPACES INDEX has the ALL indicator, the produced statement EXCLUDE INDEXSPACES INDEX will also have the ALL indicator.

Requirements for LISTDEF command syntax

The Db2 Cloning Tool Table Space Cloning LISTDEF syntax is similar to Db2 LISTDEF command syntax. However, Db2 Cloning Tool Table Space Cloning LISTDEF syntax does not make use of continuation characters. You must enter Db2 Cloning Tool Table Space Cloning LISTDEF keywords in a specified order. The order is as follows:

```
LISTDEF listname
INCLUDE | EXCLUDE
[ TABLESPACES | INDEXSPACES [COPY (YES | NO) ] ]
DATABASE database-name
  | STOGROUP (stogroupname | stogroupmask)
  | TABLESPACE database.table-space-name
  | INDEXSPACE database.index-space-name
  | TABLE creator-id.table-name
  | INDEX creator-id.index-name
[ PARTLEVEL( integer ) ]
[ CLONED(YES | NO) ]
[ RI ]
[ HISTORY ]
[ ALL | BASE | LOB | XML ]
```

For readability, this example shows each keyword on its own line, but that is not required; refer to the examples for variations.

Examples

```
LISTDEF ZTS0405A
  INCLUDE TABLESPACES TABLESPACE ZDSTDB04.ZDSTTS04 RI ALL
  INCLUDE TABLESPACES TABLESPACE ZDSTDB05.ZDSTTS05 RI ALL
```

```
LISTDEF ZTN
  INCLUDE TABLESPACES DATABASE ZTNDB
  INCLUDE INDEXSPACES DATABASE ZTNDB
  EXCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN7TSN1
  INCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN5TSN1 PARTLEVEL(0001)
  INCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN5TSN1 PARTLEVEL(0002)
  INCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN5TSN1 PARTLEVEL(0003)
```

LISTDEF command syntax

```
LISTDEF listname
```

Required keywords:

```
LISTDEF listname
INCLUDE | EXCLUDE
DATABASE databasename
  | STOGROUP( stogroupname | stogroupmask )
  | TABLESPACE database.tablespace-name
  | INDEXSPACE database.indexspace-name
  | TABLE creatorid.table-name
  | INDEX creatorid.index-name
optional keywords . . .
```

Optional keywords:

```
[ HISTORY ]
[ ALL | BASE | LOB | XML ]
[ CLONED(YES | NO) ]
[ PARTLEVEL( integer ) ]
[ RI ]
[ TABLESPACES | INDEXSPACES [COPY (YES | NO) ] ]
```

Note: COPY can only be specified with INDEXSPACES; YES | NO is required if COPY is specified.

LISTDEF command and keyword definitions

Required keywords are described first, followed by optional keywords.

LISTDEF *listname*

The LISTDEF command controls the selection of the table spaces and index spaces being copied from the source to the target Db2. It allows selection of individual partitions of a partitioned table space, or LOB columns, and the RI keyword causes it to select all table spaces and index spaces in an RI relationship. *listname* is the name (up to 18 alphanumeric characters in length) of the defined list.

- Default: None
- Required: Yes
- Restrictions: None

Required and Conditionally Required Keywords

INCLUDE

EXCLUDE

The INCLUDE/EXCLUDE parameters determine if the selected table spaces and index spaces are to be added or excluded from the list.

- Default: None
- Required: Yes
- Restrictions: EXCLUDEs should not be followed by an overriding INCLUDE.

DATABASE *database-name*

STOGROUP (*stogroupname* | *stogroupmask*)

TABLESPACE *database.tablespace-name*

TABLE *creator-id.table-name*

INDEXSPACE *database.indexspace-name*

INDEX *creator-id.index-name*

The STOGROUP, DATABASE, TABLESPACE, TABLE, INDEXSPACE, and INDEX parameters specify the type of object being named. The STOGROUP parameter is used to select all objects in all databases in a Db2 storage group. Internally, the LISTDEF statement is replaced by one statement for each database in the Db2 stogroup. STOGROUP % is not valid. Wildcards other than the single % are allowed. For example: INCLUDE INDEXSPACES STOGROUP AB* is allowed.

If the object type is TABLE or INDEX, and the DEFAULT-SQLID parameter is specified, only one qualifier name can be specified for an object. If the object type is TABLE or INDEX, and the DEFAULT-SQLID parameter is not specified, an error message is issued. If the DEFAULT-SQLID parameter is not provided, the *creator-id* should be explicitly specified.

- Default: None
- Required: Yes
- Restrictions: Type specification (TABLESPACES or INDEXSPACES) should be specified if object type DATABASE or STOGROUP is specified. For *database-name*, you cannot specify DSNDB01, DSNDB06, or DSNDB07.

Optional Keywords

HISTORY

The HISTORY parameter specifies that history tables associated with temporal tables are to be included in the list. When HISTORY is specified, only history tables are processed on that LISTDEF statement. If HISTORY is not specified, no history tables are processed on that LISTDEF statement. To process both history and non-history tables, specify the COPY command parameter ALWAYS-COPY-HISTORY-TABLES(Y).

Object specifications must include the table space, table or index objects, not their corresponding history objects. For example, specify INCLUDE TABLESPACES TABLE *creator.table* HISTORY ALL, not INCLUDE TABLESPACES TABLE *creator.table_hist* HISTORY ALL.

- Default: None
- Required: No
- Restrictions: Valid only for Db2 10 and later subsystems.

ALL
BASE
LOB
XML

Identifies the scope of the objects to be included or excluded in the LISTDEF. If you do not include one of these keywords, ALL is the default, except for the EXCLUDE INDEXSPACES INDEXSPACE/INDEX statements; those statements do not use the ALL option unless you specify it on the LISTDEF statement.

Note: Jobs that were created before APAR PI63601 is applied and that use LISTDEF without any of the keywords (ALL, BASE, LOB, or XML) may include or exclude additional objects in the LISTDEF, especially XML-related objects.

(Default) ALL specifies that base, LOB, and XML objects are included or excluded.

BASE specifies that only the base objects are included or excluded.

LOB specifies only LOB table spaces are included or excluded. Indexes are included or excluded for the selected spaces only.

XML specifies that only XML table spaces are included or excluded. Indexes are included or excluded for the selected spaces only.

- Default: ALL, except for the EXCLUDE INDEXSPACES INDEXSPACE/INDEX statements; those statements do not use the ALL option unless you specify it on the LISTDEF statement.
- Required: No
- Restrictions: If BASE, LOB, or XML is specified, the copied target space(s) may be unusable.

CLONED (YES | NO)

The CLONED parameter is used to selectively include or exclude Db2 clone tables.

YES specifies that only table spaces and index spaces that contain cloned objects are to be copied. The base table is always included with the clone table.

NO specifies that only table spaces and index spaces that do not contain cloned objects are to be copied. If you specify NO, base instances of objects with clones will not be included.

CLONED is optional, but YES or NO is required if CLONED is specified.

- Default: None
- Required: No
- Restrictions: None

PARTLEVEL(*integer*)

The PARTLEVEL parameter is used to specify individual partitions to be processed. When using PARTLEVEL for table spaces and index spaces, you must ensure that there is no update activity on partitions that are not included in LISTDEF. Failure to do this may result in a partitioned space where all partitions are not in sync with one another.

- Default: None
- Required: No
- Restrictions: PARTLEVEL is not valid with RI.

RI

The RI parameter specifies that referentially related table spaces and index spaces are to be included (or excluded) from the list. When RI is specified along with LISTDEF INCLUDE statements, referentially related objects are added to the scope of the LISTDEF in order to include them in the

process. When RI is specified along with LISTDEF EXCLUDE statements, referentially related objects are added to the scope of the LISTDEF in order to exclude them from processing.

- Default: Determined by the INCLUDE-ALL-RI keyword in the COPY command.
- Required: No
- Restrictions: RI is not valid with PARTLEVEL.

TABLESPACES

INDEXSPACES COPY (YES | NO)

The INDEXSPACES and TABLESPACES parameters specify whether this statement pertains to table spaces or index spaces. COPY can only be specified with INDEXSPACES and must immediately follow INDEXSPACES; YES or NO is required if COPY is specified.

- Default: None
- Required: Yes, if DATABASE or STOGROUP object type is specified.
- Restrictions: None

SET

The SET command is required.

SET specifies the local Db2 subsystem and optionally, other job-wide specifications for the source, target, TCP/IP server and report jobs.

SET command syntax

SET

Required keywords:

LOCAL-SSID(*localsubsystem*)
optional keywords . . .

Optional keywords:

```
[ ADVISORY-STATUS-VALUES( status1,status2,...statusn | blank ) ]
[ CONNECT-DB2-ON-CLIENT-CONNECT( Y | N ) ]
[ DB2-COMMAND-RESPONSE-WAIT ( nnnn ) ]
[ DB2-PLAN( planname ) ]
[ DEFAULT-SQLID( sqlid ) ]
[ INTELLIGENT-REBUILD( Y | N ) ]
[ IP-VERSION6( Y | N ) ]
[ KEEP-DATABASES-ON-DISCONNECT ( Y | N ) ]
[ MAX-COPY-RC( 0 | 4 | 8 ) ]
[ MAX-RC( 0 | 4 ) ]
[ MAX-SUBTASKS( number of subtasks ) ]
[ MERGE-PRINT( Y | N ) ]
[ PROCESS-UNMATCHED-TARGET-INDEXES ( Y | N ) ]
[ REBUILD-INDEXES-EXECUTE( Y | N ) ]
[ REBUILD-INDEXES-REPORT ( Y | N ) ]
[ REBUILD-INDEXES-SORTDEVT( SYSDA ) ]
[ REBUILD-UNMATCHED-TARGET-INDEXES ( Y | N ) ]
[ REMOTE-CONNECT-TYPE( C | T ) ]
[ REPORT-JOB( Y | N ) ]
[ RESTRICT-STATUS-VALUES( status1,status2,...statusn | blank ) ]
[ SCAN-ONLY( Y | N ) ]
[ SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT ( Y | N ) ]
[ SOURCE-IP-VERSION6 ( Y | N ) ]
[ SOURCE-TCP-SERVER-JOB ( Y | N ) ]
[ SOURCE-TCPIP-SERVER-IP ( ip ) ]
[ SOURCE-TCPIP-SERVER-NAME ( name ) ]
[ SOURCE-TCPIP-SERVER-PORT ( port ) ]
[ SOURCE-TCPIP-STC-NAME ( started task name ) ]
[ SUBTASK-DATASET-EXTENSIONS( Y | N ) ]
[ SUBTASK-TERMINATION-WAIT ( nnnn ) ]
[ TARGET-JOB( Y | N ) ]
```

```
[ TARGET-JOB-MAX-RC( 0 | 4 | S | P ) ]
[ TARGET-JOB-REPAIR-EXECUTE( Y | N ) ]
[ TCP-SERVER-JOB( Y | N ) ]
[ TCPIP-SERVER-IP( ip ) ]
[ TCPIP-SERVER-NAME( name ) ]
[ TCPIP-SERVER-PORT( port ) ]
[ TCPIP-STC-NAME( started task name ) ]
[ TEMPLATE-VARIABLE( variablename1,variablestring1,...
, variablenamen,variablestringn ] ) ]
[ UPDATE-DOCID-JCL-DSN( dsname ) ]
[ USE-RUNTIME-REPOSITORY( Y | N ) ]
[ UTILITY-COMMAND-EXECUTE-PERCENT(percentage per call) ]
```

SET command and keyword definitions

Required and conditionally required keywords are described first, followed by optional keywords.

SET

The SET command is required. It specifies the source Db2 subsystem and, optionally, other job-wide specifications for each of the tables space cloning jobs (source, target, TCP/IP server, and source TCP/IP server).

- Default: None
- Required: Yes
- Restrictions: None

Required SET keywords

Note: The Jobs entry in this parameter list displays the jobs in which you are allowed to enter the parameter with a SET command.

LOCAL-SSID(*localsubsystem*)

The LOCAL-SSID parameter provides the name of the local Db2 subsystem. This is the source subsystem for the source and source TCP/IP server jobs and the target subsystem for the target and target TCP/IP server jobs.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: LSSID
- Jobs: Source, source TCP/IP server, target TCP/IP server

Optional SET keywords

ADVISORY-STATUS-VALUES (*status1,status2,...statusn* | blank)

The ADVISORY-STATUS-VALUES parameter allows you to specify which status values are to be checked for table spaces and index spaces before the copies are performed. When a specified status is detected, the space is marked as mismatched and a warning message is issued. If ALLOW-COPY-ON-MISMATCH and MAX-RC(4) are in effect, the copy may proceed. Otherwise, the copy is not allowed for all affected data sets.

This keyword determines the content of the Db2 DISPLAY command:

```
-DISPLAY DATABASE(dbname) SPACENAM(*) ADVISORY(s1,s2,...sn)
```

where *dbname* is one of the databases that is being copied and *s1,s2,...,sn* are valid status values.

The following are valid values for *status1...statusn*:

- ICOPY: informational COPY-pending
- AUXW: auxiliary warning
- ARBDP: advisory REBUILD-pending

- AREO*: advisory REORG-pending
- AREOR: advisory REORG-pending
- Blank: If SET ADVISORY-STATUS-VALUES() is used, the ADVISORY portion of the DISPLAY command is turned off.

One DISPLAY command is generated for each database that has one or more spaces that are to be copied. Both the source and target subsystems are checked for the specified status values. If a default value or a specified status value does not apply to the Db2 version, it is deleted from the command.

The behavior of ADVISORY-STATUS-VALUES changes if DDL-ENABLE(Y) is specified. When these parameters are specified, AREO* and AREOR are not treated as errors, as AREO* and AREOR are expected in that situation. To treat AREO* and AREOR as errors, specify ADVISORY-STATUS-VALUES and explicitly include AREO* or AREOR in the list of values.

- Default: ARBDP, AREO*, AREOR, and AUXW are used if no SET ADVISORY-STATUS-VALUES command is found.
- Required: No
- Restrictions: None
- Short form: ADVIS
- Jobs: Source



CAUTION: If objects are that have been migrated from Db2 V7 have had columns added using ALTER TABLE ADD COLUMN, the AREO* status may not be displayed. In this case, you must REORG the table spaces before using Db2 Cloning Tool Table Space Cloning to copy them. Failure to perform this REORG may result in inaccessible target data.

CONNECT-DB2-ON-CLIENT-CONNECT(Y | N)

The CONNECT-DB2-ON-CLIENT-CONNECT parameter can be used to allow the TCP/IP server job to connect to Db2 only when a client connect occurs. When this parameter is set to Y, the TCP/IP server disconnects from Db2 when the client disconnects. This allows the TCP/IP server to be run as a started task; the TCP/IP server job can be started at IPL and left running. If a Db2 connect error occurs when the source job is running the TCP/IP server job, the TCP/IP server job ends with a return code of 8, and must be restarted.

- Default: N
- Required: No
- Restrictions: Specify in the target TCP/IP server job
- Short form: CDOCC
- Jobs: Target TCP/IP server

DB2-COMMAND-RESPONSE-WAIT (nnnn)

The DB2-COMMAND-RESPONSE-WAIT is the overall number of seconds (including retries) that Db2 Cloning Tool Table Space Cloning waits for a space to go from STOPP to STOP status after the STOP DB2 command is issued. The range is up to 999999 seconds.

The parameter value that you specify is used as an overall wait time. One retry is always attempted. The wait time before the first retry is 1 second. Db2 Cloning Tool Table Space Cloning has a default of 10 retries to check for the required status, with a default wait time between retries of 10 seconds, for a total overall wait time of 100 seconds. If the parameter value that you provide is larger than 100 seconds, then the specified value is divided by 10 (the default number of retries) and the resulting time is used as the wait time between retries. If the parameter value that you provide is less than 100 seconds, 10 seconds (the default wait time) is used as the wait time between retries. After each call, the wait time is subtracted from the general wait time (for the first call, the wait time is 1 second; for subsequent calls, the value is calculated as described). Db2 Cloning Tool Table Space Cloning proceeds as follows:

- If the result of the subtraction is a negative value, waiting stops. If the object is not in the desired status, an error is issued.

- If the result of subtraction is a positive value, then a retry is attempted.

This process repeats until the overall wait time (the specified parameter value) is exceeded, which stops the wait.

- Default: 60 seconds.
- Required: No
- Restrictions: None
- Short form: DB2CWAIT
- Jobs: Source, TCP/IP server

DB2-PLAN(*plan_name*)

The DB2-PLAN parameter specifies the Db2 plan name for source and target Db2 subsystem connects.

Note: The target job inherits the SET value from the source job.

- Default: The value that is entered in the PARMLIB, or CKZPLAN if no PARMLIB value is provided.
- Required: No
- Restrictions: None
- Short form: PLAN
- Jobs: Source, TCP/IP server

DEFAULT-SQLID(*sqlid*)

The DEFAULT-SQLID parameter value is used when a TABLECREATOR value is not specified for a MASKDEF rule, and when LISTDEF statements with an object type of TABLE or INDEX have only one object qualifier.

- Default: None
- Required: No, unless you want to specify a MASKDEF rule without using the TABLECREATOR keyword, or LISTDEF statements with object type TABLE or INDEX have only one object qualifier.
- Restrictions: None
- Short form: DEFSQLID
- Jobs: Source

INTELLIGENT-REBUILD(Y | N)

Note: This parameter will always be set to Y, regardless of the value that is specified in JCL.

This keyword generates and runs REBUILD for indexes that are related to a given table space in one REBUILD statement. REBUILD commands are generated for all indexes (or index partitions) included in the COPY. The indexes are grouped by table space to leverage multi-tasking capabilities built into REBUILD. For each table space, indexes and index partitions that require REBUILD are rebuilt in one REBUILD statement (as long as the indexes were copied successfully and the table space was copied successfully in the same COPY statement).

Most indexes can be included in a single REBUILD statement. However, if you have indexes on XML columns or clone tables, then Db2 Cloning Tool separates the REBUILD statements for the same table space, as follows:

- REBUILD for all indexes without XML columns for base tables
- REBUILD for all indexes without XML columns for clone tables
- REBUILD for all indexes with XML columns for base tables
- REBUILD for all indexes with XML columns for clone tables

An optional DD with DD name CKZINTRB can be supplied in the target job JCL. If this DD is included, a JCL version of the INTELLIGENT-REBUILD stream is written to the DD file. The JCL job can be used with modifications if a restart should be required. The DD that is supplied must be LRECL=80 and

RECFM=FB. To modify the JCL for a restart, change the job card and //STEPLIB DD as appropriate, remove the index rebuild steps that were successful, edit the step that failed (if needed), and submit.

If you also specify REBUILD-UNMATCHED-TARGET-INDEXES(Y), unmatched indexes (indexes that are only on the target) are rebuilt with the other indexes for that table space during intelligent rebuild processing. If the optional CKZINTRB DD is present, unmatched indexes are also included in the CKZINTRB JCL stream.

- Default: Y
- Required: No
- Restrictions: None
- Short form: INTRB
- Jobs: Source

IP-VERSION6 (Y | N)

The IP-VERSION6 parameter specifies the version of TCP/IP used on the TCP/IP server. N (the default) specifies IPv4. Y specifies IPv6. If you specify IP-VERSION6 in the source job, it is ignored.

- Default: N
- Required: No
- Restrictions: None
- Short form: IPV6
- Jobs: Target TCP/IP server

KEEP-DATABASES-ON-DISCONNECT (Y | N)

When using a TCP/IP server, KEEP-DATABASES-ON-DISCONNECT (Y) can be used to populate the server with prefetched target objects. Populate the server cache of database objects by running source jobs using ENABLE-TARGET-PREFETCH(Y) and PGM(NONE). The TCP/IP server job saves objects as long as it has enough storage to hold them. If the cache does not need to be refreshed, use the READ-FROM-SERVER-CACHE command with ENABLE-TARGET-PREFETCH(N) to read objects from the populated cache. If a target object changes before the source COPY job is run, rerun the source job using ENABLE-TARGET-PREFETCH(Y) and PGM(NONE) to clean and populate the cache with updated data. When using the TARGET-PREFETCH-DATABASE-LIST, every object in the list of databases is cached; otherwise, the list of databases from LISTDEF will be used.

- Default: N
- Required: No
- Restrictions: None
- Short form: KDBOD
- Jobs: TCP/IP server

MAX-COPY-RC (0 | 4 | 8)

Specifies the maximum return code for data set copy. When the specified return code is exceeded, the job ends in error. Use this parameter in the source job to override the default or PARMLIB specification. This parameter allows copies to continue even with warnings (value 4) or errors (value 8). Thus, all copies that can complete while all spaces are stopped completes.

- Default: PARMLIB specification for MAX_COPY_RC. If no value is provided in PARMLIB, the default is 8.
- Required: No
- Restrictions: None
- Short form: MXCPRC
- Jobs: Source

MAX-RC (0 | 4)

Specifies the maximum job return code. When the specified return code is exceeded, the job ends in error. Use this parameter in the source or TCP/IP jobs to override the default or PARMLIB specification. This parameter determines whether the job continues with warnings or terminates.

- Default: PARMLIB specification for MAX_RC. If no value is provided in PARMLIB, the default is 4.
- Required: No
- Restrictions: None
- Short form: MXRC
- Jobs: Source, source TCP/IP server, target TCP/IP server

MAX-SUBTASKS (number of subtasks)

Specifies the number of subtasks to start. Use this parameter in the source job to override the default or PARMLIB specification. This parameter can be used to reduce the elapsed time of long-running source and target jobs by allowing multiple I/O operations to run concurrently. Db2 processing is impacted in the source job and VSAM processing in the target job. As the target job must access every page, it is particularly useful. Refer to the MAX_SUBTASKS description in the topic [“CKZINI keyword syntax and descriptions”](#) on page 1146 for more recommendations for the value of this parameter.

Note: The target job inherits the SET value from the source job. To use different values for each job, use two PARMLIB members: one for the source job and one for the target job, each with its own MAX_SUBTASKS value.

When using a high value for subtasks, specify SUBTASK-TERMINATION-WAIT(*nnn*) in the source job. *nnn* should be enough seconds to allow all subtasks to come down. If it is not large enough, an A03 abend occurs. The SUBTASK-TERMINATION-WAIT command will also be used in the target job.

- Default: PARMLIB specification for MAX_SUBTASKS. If no value is provided in PARMLIB, the default is 1.
- Required: No
- Restrictions: Valid values are from 1 - 99.
- Short form: MXSUBT
- Jobs: Source

MERGE-PRINT (Y | N)

This parameter allows message output from CKZPRINT and CKZLOG to be combined into CKZPRINT. It can be specified in the source or TCP/IP server jobs and is useful when you are investigating a problem or sending documentation to IBM Software Support.

Note: The target job inherits the SET value from the source job.

- Default: N
- Required: No
- Restrictions: None
- Short form: MERGEP
- Jobs: Source, source TCP/IP server, target TCP/IP server

PROCESS-UNMATCHED-TARGET-INDEXES (Y | N)

This keyword causes the target job to search for, and report on, any unmatched (target-only) indexes. An unmatched index is an index on the target for which there is no corresponding index on the source.

- Default: N
- Required: When REBUILD-UNMATCHED-TARGET-INDEXES(Y) is used
- Restrictions: None
- Short form: PUTIX
- Jobs: Source

REBUILD-INDEXES-EXECUTE (Y | N)

When set to Y, the REBUILD-INDEXES-EXECUTE parameter indicates that index rebuilds are to be submitted as part of the target job.

- Default: Y
- Required: No
- Restrictions: None
- Short form: RBIXX
- Jobs: Source

REBUILD-INDEXES-REPORT (Y | N)

When set to Y, REBUILD-INDEXES-REPORT provides a concise and easily readable summary of indexes for which REBUILD was generated, by table space.

When INTELLIGENT-REBUILD is set to Y, detail lines are grouped by table space. For each table space, index spaces are listed as follows:

- Index spaces, index space partitions, and NPI/NPSI/DPSI logical partitions
- If REBUILD-UNMATCHED-TARGET-INDEXES(Y), unmatched index spaces are listed with the corresponding table space

The report is displayed only in the target job and contains the following information:

```
CKZ55502I - Begin REBUILD INDEXES report

                S   C   P
                PT  L   A
                AY  O   R
                CP  N   E
                EE  E   D
DBNAME         SNAME      RC   TS   B   PART   UNMATCHED
-----
DBNAME01      TSNAME01    0   TS   B
DBNAME01      TSNAME01    0   TS   C
DBNAME02      ISNAME01    0   IS   +   *   +
DBNAME03      ISNAME03    0   IS   +   1
DBNAME04      ISNAME02    0   IS           4

CKZ55503I - End REBUILD INDEXES report
```

- Default: N
- Required: No
- Restrictions: Required REBUILD-INDEXES-EXECUTE(Y)
- Short form: RBIRP
- Jobs: Source

REBUILD-INDEXES-SORTDEVT (SYSDA)

When specified, the value of REBUILD-INDEXES-SORTDEVT is used as the value for the SORTDEVT parameter for the REBUILD INDEX utility and specifies the device type for temporary data sets that are to be dynamically allocated by the sort program. If the REBUILD-INDEXES-SORTDEVT keyword is not specified in the source job, then REBUILD-INDEXES-SORTDEVT(SYSDA) is used by default. For the device type, you can specify any disk device that is valid on the DYNALLOC parameter of the SORT or OPTION options for the sort program (however, tape devices are not supported by the sort program).

- Default: SYSDA
- Required: No
- Restrictions: None
- Short form: RBISDT
- Jobs: Source

REBUILD-UNMATCHED-TARGET-INDEXES (Y | N)

Unmatched indexes are target-only indexes for tables included in the COPY. This keyword allows automatic rebuild of those indexes after a successful copy of a table space.

When used with INTELLIGENT-REBUILD(Y):

- Index partitions that require rebuild and were copied successfully are rebuilt in one REBUILD statement, if the table space was copied successfully in the same COPY statement.
- Nonpartitioned secondary indexes (NPIs or NPSIs) and data-partitioned secondary indexes (DPSIs) that require rebuild and were copied successfully are rebuilt in one REBUILD statement, if the table space was copied successfully in the same COPY statement.
- Default: N
- Required: No
- Restrictions: Required REBUILD-INDEXES-EXECUTE(Y) and PROCESS-UNMATCHED-TARGET-INDEXES(Y)
- Short form: RUTIX
- Jobs: Source

REMOTE-CONNECT-TYPE (C | T)

The REMOTE-CONNECT-TYPE parameter is used to specify a particular connection type for the remote (target) subsystem. Specify C to indicate CAF or T for TCP/IP.

If the keyword is not included, connections are attempted in this order: CAF and TCP/IP. This command is the preferred method of specifying TCP/IP connections. For example, if no keyword is specified when TCP/IP is used and CAF fails, error messages are printed for the CAF failure. If SET REMOTE-CONNECT-TYPE(T), then CAF is not attempted and thus no error message will be issued.

- Default: If this keyword is not specified, connections are attempted in this order: C, T.
- Required: No
- Restrictions: None
- Short form: RCTYP
- Jobs: Source

REPORT-JOB (Y | N)

This parameter is used to determine whether a report based on data in the target job runtime repository is to be output. The report job runs as a separate job. A sample job with instructions is contained in SCKZJCL (CKZREPJB).

- Default: N
- Required: No
- Restrictions: None.
- Short form: REPJOB
- Jobs: Report

RESTRICT-STATUS-VALUES (*status1,status2,...statusn* | blank)

The RESTRICT-STATUS-VALUES parameter allows you to specify which status values are to be checked for table spaces and index spaces before the copies are performed. When a specified status is detected, the space is marked as mismatched and a warning message is issued. If ALLOW-COPY-ON-MISMATCH and MAX-RC(4) are in effect, the copy proceeds. Otherwise, the copy is not allowed for all affected data sets.

This keyword determines the content of the Db2 DISPLAY command:

```
-DISPLAY DATABASE(dbname) SPACENAM(*) RESTRICT(s1,s2,...sn)
```

where *dbname* is one of the databases that is being copied and *s1,s2,...,sn* are valid status values.

The following are valid values for *status1...statusn*:

- ACHKP: auxiliary CHECK-pending
- CHKP: CHECK-pending
- COPY: COPY-pending
- GRECP: group buffer pool RECOVER-pending
- LPL: logical page list entries
- RBDP: REBUILD-pending
- RECP: RECOVER-pending
- REORP: REORG-pending
- RO: Read-only mode
- STOP: Stopped objects, including the restricted states STOP, STOPE, STOPP, and LSTOP
- UT: Utility access mode
- UTRO: Serialized for utility access and available for read-only access
- UTRW: Serialized for utility access and available for read-write access
- UTUT: Serialized for utility access and unavailable
- UT*: Any utility access mode: UT, UTRW, UTRO, or UTUT
- Blank: If SET RESTRICT-STATUS-VALUES() is used, the RESTRICT portion of the DISPLAY command is turned off.

One DISPLAY command is generated for each database that has one or more spaces that are to be copied. Both the source and target subsystems are checked for the specified status values. If a default value or a specified status value does not apply to the Db2 version, it is deleted from the command.

For statuses RO, STOPP, and LSTOP, no warning messages are issued.

- Default:
 - If no SET RESTRICT-STATUS-VALUES command is provided, the defaults are ACHKP, CHKP, GRECP, RBDP, RECP, REORP, and UT*. In addition to those defaults, if LOG-APPLY is enabled with QUIESCE-POINT(Y), COPY is added to the list of default values for the DISPLAY command on the source subsystem.
 - If the SET RESTRICT-STATUS-VALUES command has an empty value () or blank value () and LOG-APPLY is enabled with QUIESCE-POINT(Y), the default values are ACHKP, CHKP, COPY, RBDP, and RECP for the DISPLAY command on the source subsystem.
- Required: No
- Restrictions: None
- Short form: RESTR
- Jobs: Source

SCAN-ONLY(Y | N)

The SCAN-ONLY parameter controls the target job updating OBIDs and LOGRBAs in the target job.

If Y, then all the target data sets are read, and page change messages are output in the log; however the page sets are not written.

- Default: N
- Required: No
- Restrictions: This parm is set to Y by Db2 Cloning Tool Table Space Cloning when the CKZSYNC DD is updated in the source job during a simulation (SIMULATE(A)) run. This parm is set to N by Db2 Cloning Tool Table Space Cloning when the CKZSYNC DD is updated in the source job during a non-simulation (SIMULATE(N)) run. Before running the target job, this parm may be changed manually to Y in the CKZSYNC DD as long as no SIMULATE(A) restricted parameters are enabled for the target job (DATA-MASKING(Y), SUBTASK-DATASET-EXTENSIONS(Y), or USE-RUNTIME-REPOSITORY(Y)).

- Short form: SCANO
- Jobs: None. Do not specify or change this parameter yourself, unless the target job is needed to run in simulate mode AND no SIM(A) restricted parameters are specified for the target job.

SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT (Y | N)

The SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT parameter can be used to allow the source TCP/IP server job to connect to Db2 only when a client connect occurs. When this parameter is set to Y, the source TCP/IP server disconnects from Db2 when the client disconnects. This allows the source TCP/IP server to be run as a started task; the source TCP/IP server job can be started at IPL and left running. If a Db2 connect error occurs when the source job is running the source TCP/IP server job, the source TCP/IP server job ends with a return code of 8, and must be restarted.

- Default: N
- Required: No
- Restrictions: Specify in the source TCP/IP server job
- Short form: SCDOCC
- Jobs: Source TCP/IP server

SOURCE-IP-VERSION6 (Y | N)

The SOURCE-IP-VERSION6 parameter specifies the version of TCP/IP used on the source TCP/IP server. N (the default) specifies IPv4. Y specifies IPv6.

- Default: N
- Required: No
- Restrictions: Specify in the source TCP/IP server job
- Short form: SIPV6
- Jobs: Source TCP/IP server

SOURCE-TCP-SERVER-JOB (Y | N)

This parameter distinguishes the source TCP/IP server job from the source job and target job.

- Default: N
- Required: No
- Restrictions: Specify in the source TCP/IP server job
- Short form: STCPJOB
- Jobs: Source TCP/IP server

SOURCE-TCPIP-SERVER-IP (ip)

This parameter specifies the connection IP address for the LPAR on which the source TCP/IP server is executing.

- Default: None
- Required: No
- Restrictions: Specify in the source job. If both SOURCE-TCPIP-SERVER-NAME and SOURCE-TCPIP-SERVER-IP are specified, SOURCE-TCPIP-SERVER-NAME takes precedence over SOURCE-TCPIP-SERVER-IP.
- Short form: SSIP
- Jobs: Source

SOURCE-TCPIP-SERVER-NAME (name)

This parameter specifies the connection DNS name for the LPAR on which the source TCP/IP server is running.

- Default: None
- Required: No

- Restrictions: Specify in the source job. If both SOURCE-TCPIP-SERVER-NAME and SOURCE-TCPIP-SERVER-IP are specified, SOURCE-TCPIP-SERVER-NAME takes precedence over SOURCE-TCPIP-SERVER-IP.
- Short form: SSRVNAME
- Jobs: Source TCP/IP server

SOURCE-TCPIP-SERVER-PORT (*port*)

This parameter allows you to specify the connection port that the source TCP/IP server is to use. Use this parameter to override the default and/or PARMLIB specification. Valid values are 1 to 65535.

- Default: PARMLIB specification for TCPIP_SERVER_PORT, or 5099 if no PARMLIB value is provided.
- Required: Yes; in the source TCP/IP server job.
- Restrictions: Specify in the source or source TCP/IP server job
- Short form: STSPORT
- Jobs: Source, source TCP/IP server

SOURCE-TCPIP-STC-NAME (*started task name*)

This parameter allows you to specify the name of the TCP/IP address space that is used for communications between the target job and the source TCP/IP server job. This parameter can be specified in the source and the source TCP/IP server jobs to override the default or PARMLIB specification. When specified in the source job, this parameter is passed to the target job and should be set to the name of the TCP/IP address space on the target LPAR. When this parameter is specified in the source TCP/IP server job, it is used by the TCP/IP server job itself and should specify the name of the TCP/IP address space on the source LPAR.

- Default: For the source job, the default is the target job PARMLIB specification for TCPIP_STC_NAME, or TCPIP if no PARMLIB value is provided. For the source TCP/IP server job, the default is the PARMLIB specification for TCPIP_STC_NAME, or TCPIP if no PARMLIB value is provided.
- Required: No
- Restrictions: Specify in the source or source TCP/IP server jobs
- Short form: STSNAME
- Jobs: Source, source TCP/IP server

SUBTASK-DATASET-EXTENSIONS(*Y | N*)

The SUBTASK-DATASET-EXTENSIONS parameter allows data set extents to be processed in any available subtask. When a significant number of extents may be processed by the target job, setting this parameter to Y might improve the total elapsed time of the target job. If Db2 START commands are enabled for target objects, the objects will be started when the last extent of an object has been processed, regardless of the subtask used. The number of subtasks (set via PARMLIB parameter MAX_SUBTASKS or COPY command MAX-SUBTASKS parameter) must be greater than one for this to be a useful feature.

- Default: N
- Required: No
- Restrictions: Not allowed with PGM(SRCIMCPY), PGM(SRCVSCP), when using data masking or log apply functionality, or when Db2 STOP commands may appear in the target job (refer to the COPY parameter AUTO-STOP-TARGET-SPACE or PARMLIB parameter AUTO_STOP_TARGET_SPACE).
- Short form: SUBDX
- Jobs: Source

SUBTASK-TERMINATION-WAIT(*nnnn*)

The SUBTASK-TERMINATION-WAIT keyword allows you to specify a wait time for all subtasks to end before exiting. In the target job, a large data set being updated with new OBIDs can take a long time to process. In this case, Db2 Cloning Tool Table Space Cloning may exit before the subtask ends,

resulting in an A03 ABEND. Valid values are 0 to 9999 seconds; 0 indicates that Db2 Cloning Tool Table Space Cloning is to wait for all subtasks to end before exiting.

- Default: 0
- Required: No
- Restrictions: None
- Short form: STWAIT
- Jobs: Source, target, TCP/IP server

TARGET-JOB(Y | N)

The TARGET-JOB parameter distinguishes the target job from the source job and the TCP/IP server job. This parm is set to Y by Db2 Cloning Tool Table Space Cloning when the CKZSYNC DD is updated in the source job.

The parm value is N (the default) if the source job or TCP/IP server job, and Y if the target job.

- Default: N
- Required: No
- Restrictions: Do not specify or change this parameter. It is an internal parameter that is set by Db2 Cloning Tool Table Space Cloning.
- Short form: TRGJOB
- Jobs: None.

TARGET-JOB-MAX-RC(0 | 4 | S | P)

The TARGET-JOB-MAX-RC parameter specifies the maximum target job return code. When the specified return code is exceeded, the target job ends in error. Use this parameter in the source job to override PARMLIB specification in the target job. This parameter determines whether the target job continues with warnings or terminates. Specify 0 to use maximum return code greater than 0 in the target job. Specify 4 to use maximum return code greater than 4 in the target job. Specify S to use the maximum return code from the source job in the target job. Specify P to use the maximum return code from the CKZINI PARMLIB in the target job.

- Default: P
- Required: No
- Restrictions: None
- Short form: TJMXRC
- Jobs: Source

TARGET-JOB-REPAIR-EXECUTE(Y | N)

This parameter is used when generating REPAIR utility jobs with job templates. Specify TARGET-JOB-REPAIR(Y) to generate REPAIR jobs to detect and correct catalog or Db2 version inconsistencies. This can occur when Db2 objects are copied from one subsystem to another. When TARGET-JOB-REPAIR-EXECUTE is set to Y, REPAIR jobs are submitted by Db2 Cloning Tool Table Space Cloning near the end of the target job. If the parameter is set to N or not included, REPAIR jobs are not submitted.

The inconsistencies processed are record format (BRF vs. RRF), actual page format vs. catalog column RBA_FORMAT, HASHDATAPAGES, and versioned objects. Record format, page format, and HASHDATAPAGES processing only apply to Db2 V11 and Db2 12, and may be run only on table spaces. Versioned object processing applies to Db2 V11 and Db2 12, and REPAIR CATALOG can be run only on table spaces.

- Default: Y
- Required: No
- Restrictions: None
- Short form: TRGRX
- Jobs: Source.

TCP-SERVER-JOB (Y | N)

The TCP-SERVER-JOB parameter distinguishes the target TCP/IP server job from the source job and the target job. Set this parameter to N (the default) for the source job or target job. Set this parameter to Y for the target TCP/IP server job.

- Default: N
- Required: No
- Restrictions: None
- Short form: TCPJOB
- Jobs: Target TCP/IP server

TCPIP-SERVER-IP (ip)

This parameter specifies the IP of the LPAR on which the TCP/IP server job is running.

- Default: None
- Required: No
- Restrictions: Specify in the source job. If both TCPIP-SERVER-NAME and TCPIP-SERVER-IP are specified, TCPIP-SERVER-NAME takes precedence over TCPIP-SERVER-IP.
- Short form: SIP
- Jobs: Source

TCPIP-SERVER-NAME (name)

This parameter specifies the connection DNS name for the LPAR on which the target TCP/IP server is running.

- Default: None
- Required: No
- Restrictions: Specify in the source job. If both TCPIP-SERVER-NAME and TCPIP-SERVER-IP are specified, TCPIP-SERVER-NAME takes precedence over TCPIP-SERVER-IP.
- Short form: TSRVNAME
- Jobs: Source, TCP/IP server

TCPIP-SERVER-PORT (port)

This parameter allows you to specify the port that the TCP/IP server is to use to wait for requests from the source job. It may also be used in the source job. Use this parameter to override the default and/or PARMLIB specification. Valid values are 1 to 65535.

Note: The source job port specified with this SET command shall be overridden if specified using SERVER-PORT on the COPY command. This port and the one specified for the TCP/IP server must match.

- Default: PARMLIB specification for TCPIP_SERVER_PORT, or 5099 if no PARMLIB value is provided.
- Required: Yes, in the target TCP/IP server job.
- Restrictions: None
- Short form: TSPORT
- Jobs: Source, target TCP/IP server

TCPIP-STC-NAME (started task name)

This parameter allows you to specify the name of the TCP/IP address space that is used for communications between the source job and the TCP/IP target server job. This parameter can be specified in the source and the target TCP/IP server jobs to override the default or PARMLIB specification. When specified in the source job, this parameter is used for establishing connection to the target TCP/IP server job and should be set to the name of the TCP/IP address space on the source LPAR. When this parameter is specified in the target TCP/IP server job, it used by the TCP/IP server job itself and should specify the name of the TCP/IP address space on the target LPAR.

- Default: PARMLIB specification for TCPIP_STC_NAME, or TCPIP if no PARMLIB value is provided.

- Required: No
- Restrictions: None
- Short form: TMVAR
- Jobs: Source, target TCP/IP server

TEMPLATE-VARIABLE (*variablename1,variablestring1, ... [,variablenamen,variablestringn]*)

The TEMPLATE-VARIABLE parameter allows a variable to be set for the life of the source job. The maximum variable name length is 8 characters and the maximum variable length is 44. When making a substitution, the variable expansion cannot extend the line past column 71. These are straight substitution variables, i.e., no processing is controlled by these values.

The variable name must begin with && and the variable string cannot have embedded blanks, an ampersand or field terminator (.). With SET VARIABLES, multiple copy commands to multiple targets can be built with a single source job. For example, each template could have a separate temporary variable for the high level qualifier following the RENAMEU command. If copied target data sets must be accessible, subsequent source jobs can be run against secondary targets without copying the data [PGM(NONE)], to output a SYNCDB2 file to use for target jobs to correct OBIDs.

- Default: N
- Required: No
- Restrictions: None
- Short form: TMVAR
- Jobs: Source

UPDATE-DOCID-JCL-DSN (*dsname*)

Specify this parameter when the target is a member of a data sharing group and XML table spaces are being cloned. This parameter specifies the data set name that will contain additional update jobs for target data sharing group members that must be run after the target job completes. This data set must be a PDS with RECFM=FB and LRECL=80, and must be accessible for the target job. If this keyword is not specified when cloning XML table spaces and the target is part of a data sharing group, a warning message is issued. For more information about this keyword, see [“XML considerations” on page 185](#).

- Default: None
- Required: No, unless XML table spaces are being cloned and the target is part of a data sharing group.
- Restrictions: None.
- Short form: UPDIJD
- Jobs: Source

USE-RUNTIME-REPOSITORY(*Y | N*)

This parameter is used to determine if all the phases in the target job are processed. It utilizes a target job runtime repository that keeps track of target jobs and all the data sets that are processed by the target jobs. This repository can allow the failed target job to be restarted and skip any target phases that have been successfully processed. For details about using the runtime repository, see [“Runtime repository functionality overview” on page 216](#).

For USE-RUNTIME-REPOSITORY(Y), you must:

1. Add these DDs to the target job:
 - //CKZRRJOB DD DISP=OLD,DSN=&h1q.RRJOB
 - //CKZRRDSN DD DISP=OLD,DSN=&h1q.RRDSN
2. Allocate the DDs; allocation instructions are provided in SCKZJCL(CKZRREP).
3. Add USE-RUNTIME-REPOSITORY(Y) to the SET command in the source job.

- Default: N
- Required: No
- Restrictions: USE-RUNTIME-REPOSITORY(Y) is not valid with SIM(A).

- Short form: USERR
- Jobs: Source

UTILITY-COMMAND-EXECUTE-PERCENT*(percentage per call)*

This parameter specifies the percentage of objects or data sets that are eligible for a particular Db2 utility to be run in a single call to that utility. Specify a percentage of the total data sets or objects from 1 - 50 inclusive. For example, if UTILITY-COMMAND-EXECUTE-PERCENT is set to 20, and there are 500 table spaces eligible for a QUIESCE, then five QUIESCE calls would be submitted with 100 table spaces in each call. The default specifies one utility call for all eligible spaces.

Note: Currently, the only utility this parameter can be specified for is the QUIESCE command when submitted during log apply processing.

- Default: 0 (all eligible spaces are submitted with a single call)
- Required: No
- Restrictions: None
- Short form: UTCXP
- Jobs: Source

Chapter 28. Troubleshooting

Use these topics to diagnose and correct problems that you experience with Db2 Cloning Tool.

Messages

Naming conventions

Product message identifiers are in the form pppmmnnx or pppmmnnx

where:

ppp

is the 3 alpha character product code – CKZ for Db2 Cloning Tool

mmm

is the module identifier

nn or nnn

is the message number and

x

is the message type indicator

E

error

I

information

W

warning

Example 1:

The message identifier **CKZ02056I** would be a message from module **CKZ00020**, message number of **56**, and is an **I**nformation only message.

Example 2:

The message identifier **CKZVSE11E** would be a message from module **CKZ01VSE**, message number of **11**, and is an **E**rror type message.

Messages and descriptions

Tools Customizer messages

Use the information in these messages to help you diagnose and solve Tools Customizer problems.

CCQB000I **The product parameter data was saved in the data store.**

Explanation:

Changes that were made to the product parameters were saved in the data store.

System action:

None.

User response:

No action is required.

CCQB001I **The DB2 parameter data was saved in the data store.**

Explanation:

Changes that were made to the Db2 parameters were saved in the data store.

System action:

None.

User response:

No action is required.

CCQB002I **The LPAR parameter data was saved in the data store.**

Explanation:

Changes that were made to the LPAR parameters were saved in the data store.

System action:

None.

User response:

No action is required.

CCQB003E **At least one step must be selected in a selected task. The selected task is *task_description*.**

Explanation:

When a task is selected, at least one step must be selected. A selected step is missing from the specified task.

System action:

Processing stops.

User response:

Select a step in the specified task or deselect the task.

CCQB004I **The required information to run the Discover EXEC was saved in the data store.**

Explanation:

The data store contains all the information that is required to run the Discover EXEC.

System action:

None.

User response:

No action is required.

CCQB005E **The conflicting values for the *parameter_name* parameter must be resolved before the information can be saved.**

Explanation:

Two values for one parameter conflict with each other, and they must be resolved to save the information.

System action:

Processing stops.

User response:

Resolve the conflicting values for the parameter.

CCQB006E **One row must be selected.**

Explanation:

One row in the table must be selected.

System action:

Processing stops.

User response:

Select one row.

CCQB007E **Only one row can be selected.**

Explanation:

Multiple rows in the table are selected, but only one row is allowed to be selected.

System action:

Processing stops.

User response:

Select only one row.

CCQC000I **The jobs have been customized on the selected DB2 entries.**

Explanation:

The jobs were customized on the Db2 entries that were selected.

System action:

None.

User response:

Press Enter to clear the message.

CCQC001W **The jobs were not generated on one or more of the selected DB2 entries. Press PF3 to check the Db2 entries that were not customized.**

Explanation:

The product was not customized on one or more of the Db2 entries that were selected.

System action:

None.

User response:

Press PF3 to see the Db2 entries on which the product was not customized. The status of these Db2 entries is Errors in Customization.

CCQC002I **The edit session was started automatically because values for required parameters are missing or must be verified.**

Explanation:

If product, LPAR parameters, or Db2 parameters are not defined or if parameter definitions must be verified, an editing session for the undefined or unverified parameters starts automatically.

System action:

None.

User response:

Define values for all required product, LPAR parameters, or Db2 parameters.

CCQC003W **The *template_name* template in the *library_name* metadata library does not contain any parameters.**

Explanation:

The specified template does not have parameters.

System action:

None.

User response:

No action is required.

CCQC004S **The value of the "type" attribute for the *template_name* template in the *library_name* metadata library does not match the value that was previously specified. The value is *value_name*, and the previously specified value is *value_name*.**

Explanation:

The value of the "type" attribute must match the value that was previously specified.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information" on page 1083](#). Contact IBM Software Support.

CCQC005S **The *template_name* template exceeds the number of allowed templates for a customization sequence. The template is in the *library_name* metadata library.**

Explanation:

The customization sequence can process only *number* templates. The specified template cannot be processed because the customization sequence already contains the maximum number of templates.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information" on page 1083](#). Contact IBM Software Support.

CCQC006E **The jobs could not be generated for the *group_attach_name* Db2 group attach name.**

Explanation:

The customization jobs could not be generated for the specified Db2 group attach name.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information" on page 1083](#). Contact IBM Software Support.

CCQC007E **The jobs could not be generated for the *subsystem_ID* Db2 subsystem.**

Explanation:

The customization jobs could not be generated for the specified Db2 subsystem.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information" on page 1083](#). Contact IBM Software Support.

CCQC008E **The jobs could not be generated for the *member_name* Db2 member.**

Explanation:

The customization jobs could not be generated for the specified Db2 member.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information" on page 1083](#). Contact IBM Software Support.

CCQC009S **The jobs were not generated for the DB2 entries.**

Explanation:

One or more errors occurred while customization jobs were being generated for the selected Db2 entries.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information" on page 1083](#). Contact IBM Software Support.

CCQC010S **The *template_name* template could not be accessed in the *library_name* metadata library.**

Explanation:

The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action:

Processing stops.

User response:

Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC011S **The *template_name* template could not be written to the *library_name* customization library.**

Explanation:

The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action:

Processing stops.

User response:

Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC012W The job card was generated with default values because the JOB keyword was missing.

Explanation:

Default values were used to generate the job card because the JOB keyword was not specified in the first line of the job card.

System action:

The job card was generated with default values.

User response:

No action is required. To generate the job card with your own values, add the JOB keyword in the first line of the job card.

CCQC013W The job card was generated with the default value for the programmer name because the specified programmer name exceeded 20 characters.

Explanation:

Default values were used to generate the job card because the specified programmer name contained too many characters.

System action:

The job card was generated with default values.

User response:

No action is required. To generate the job card with your own values, add a valid programmer name in the job card. A valid programmer name is 1 - 20 characters.

CCQC014W The job card was generated with default values because the JOB keyword was not followed by a space.

Explanation:

Default values were used to generate the job card because a space did not follow the JOB keyword.

System action:

The job card was generated with default values.

User response:

No action is required. To generate the job card with your own values, add a space after the JOB keyword in the job card.

CCQC015S The *template_name* template in the *library_name* metadata library contains the following file-

tailoring control statement: *statement_name*. This control statement is not valid in a *template_type* template.

Explanation:

The *template_type* template cannot contain the specified type of file-tailoring control statement.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQC016S The)DOT file-tailoring control statement exceeded the number of allowed occurrences for the *template_name* template in the *library_name* metadata library.

Explanation:

The)DOT file-tailoring control statement can occur only a limited number of times in the specified template.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQC017S The nested)DOT file-tailoring control statements exceeded the number of allowed occurrences in the *template_name* template in the *library_name* metadata library.

Explanation:

Nested)DOT file-tailoring control statements can occur only *number* times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQC018S The *template_name* template in the *library_name* metadata library is not valid because it does not contain any data.

Explanation:

The specified template is missing required data.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQC019S **The *template_name* template in the *library_name* metadata library is not valid because an)ENDDOT file-tailoring control statement is missing.**

Explanation:

A)ENDDOT file-tailoring control statement is required in the specified template.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQC021S **The *template_name* template in the *library_name* metadata library is not valid because the template must start with the *parameter_name* job card parameter.**

Explanation:

The specified template must start with the specified job card parameter.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQC022S **The parameters used in a)DOT file-tailoring control statement exceeded the number of allowed parameters in the *template_name* template. The template is in the *library_name* metadata library. The error occurs in)DOT section *section_number*.**

Explanation:

A)DOT file-tailoring control statement can contain only a limited number of parameters.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQC023S **The)DOT file-tailoring control statement must include the *table_name* table name in the *template_name* template. The template is in the *library_name* metadata library. The error occurs in)DOT section *section_number*.**

Explanation:

The)DOT file-tailoring control statement is missing a required table name.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQC024S **ISPF file tailoring failed for the *template_name* template in the *library_name* metadata library.**

Explanation:

An error occurred during ISPF file tailoring for the specified template.

System action:

Processing stops.

User response

Review the Tools Customizer-generated trace data set and the ISPF file tailoring trace data set. To create an ISPF file tailoring trace data set, complete the following steps:

1. Run Tools Customizer until the error is about to occur.
2. Specify the ISPFTRC command, and press Enter.
3. Issue the Tools Customizer command that issues the error.
4. Specify the ISPFTRC command, and press Enter. The ISPF file tailoring trace data set is created. It adheres the following naming convention: *TSO_ID*.ISPFTRACE, where *TSO_ID* is the TSO user ID that is being used.

If the problem persists, gather the following information and contact IBM Software Support.

- A screen capture of the Tools Customizer error. Ensure that the complete error message is displayed by pressing PF1.
- The Tools Customizer trace data set. It adheres to the following naming convention: *TSO_ID*.CCQTRACE, where *TSO_ID* is the TSO user ID that is running Tools Customizer.
- The ISPF file tailoring trace data set.

CCQC025I **Customized jobs do not exist because they have not been generated.**

Explanation:

The list of customized jobs cannot be displayed because the product has not been customized for any Db2 entries.

System action:

None.

User response:

Complete the steps to customize a product. Customized jobs are generated when all required product, LPAR parameters, and Db2 parameters are defined and at least one Db2 entry on which to customize the product has been selected.

CCQC026S **The value of the "customized" attribute for the *parameter_name* parameter in the *library_name* metadata library template does not match the value that was previously specified. The value is *value_name*, and the previously specified value is *value_name*.**

Explanation:

The value for the "customized" attribute for a parameter must match the value that was previously specified.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information"](#) on page 1083. Contact IBM Software Support.

CCQC027S **The *job_name* customization job was not found in the *library_name* customization library.**

Explanation:

The selected customization job does not exist in the customization library.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information"](#) on page 1083. Contact IBM Software Support.

CCQC028S **The *library_name* customization library was not found.**

Explanation:

The customization library does not exist.

System action:

Processing stops.

User response:

See ["Gathering diagnostic information"](#) on page 1083. Contact IBM Software Support.

CCQC029I **The customization jobs were generated for *Product_name*.**

Explanation:

The customization jobs were generated for the specific product.

System action:

None.

User response:

No action is required.

CCQC030S **The customization jobs cannot be generated because at least one DB2 entry must be associated with this product.**

Explanation:

The product that you are customizing requires at least one Db2 entry to be associated with it before customization jobs can be generated.

System action:

None.

User response:

Associate a Db2 entry with the product that you are customizing, and regenerate the jobs.

CCQC031I **The jobs were generated for the associated DB2 entries.**

Explanation:

The customization jobs were generated for the Db2 entries that are associated with the product.

System action:

None.

User response:

No action is required.

CCQC032S **The customization jobs were not generated for *Product_name*.**

Explanation:

A severe error occurred while the jobs were being generated for the specified product.

System action:

None.

User response:

See ["Gathering diagnostic information"](#) on page 1083. Contact IBM Software Support.

CCQC033S **The *customization_library_name* has no customized jobs.**

Explanation:

The specified customization library cannot be browsed or edited because it is empty.

System action:

None.

User response:

Generate customization jobs for the specified library, and browse or edit the library again.

CCQC034S **The specified operation is not allowed.**

Explanation:

Issuing commands against customization jobs from the customization library from an ISPF browse or edit

session that was started on the **Finish Product Customization** panel is restricted.

System action:

None.

User response:

To make changes to customization jobs, follow the steps for recustomization.

CCQC035E **Before you generate customization jobs, edit the product parameters to select one or more tasks or steps, and then issue the G line command or the GENERATEALL command again.**

Explanation:

One or more tasks or steps must be selected before customization jobs can be generated.

System action:

None.

User response:

Edit the product parameters to select one or more tasks or steps. Then, issue the G line command or the GENERATEALL command again.

CCQC036E **Before you exit the Product Parameters panel, you must select one or more tasks or steps to generate customization jobs or issue the CANCEL command.**

Explanation:

One or more tasks or steps must be selected to generate customization jobs or the CANCEL command must be issued before you can exit the **Product Parameters** panel.

System action:

None.

User response:

Select one or more tasks or steps, or issue the CANCEL command.

CCQC037W **The customization information was not found.**

Explanation:

To use the JOBLIST command, the customization jobs must be regenerated by using the GENERATEALL command or the G line command.

System action:

None.

User response:

Issue the GENERATEALL command or the G line command to generate the customization jobs.

CCQC038W **The customization information cannot be accessed because the**

customization_member customization member is being used.

Explanation:

The customization member that was specified on NNN is currently being used.

System action:

None.

User response:

Determine why the customization member is in use, release it, and redo the work.

CCQC039I **The VERIFY/VERIFYOFF command is not active in Generate mode.**

Explanation:

The VERIFY/VERIFYOFF command is not active in Generate mode because all values saved in Generate mode must already have been verified.

System action:

None.

User response:

No action is required.

CCQD000W **The member_name environment index member is not valid. The PL/I XML parser issued the following exception warning code: code_number.**

Explanation:

While determining if the specified environment index member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQD001S **The member_name environment index member is not valid. The PL/I XML parser issued the following exception error code: code_number.**

Explanation:

While determining if the specified environment index member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQD002S **The XML structure of the *member_name* environment index member is not valid. The *element_name* element is unknown.**

Explanation:

The specified environment index member contains an unknown element.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD003S **The XML structure of the *member_name* environment index member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

Content was found in an element that cannot contain content.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD004S **The XML structure of the *member_name* environment index member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element does not contain required content.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD005S **The XML structure of the *member_name* environment index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD006S **The XML structure of the *member_name* environment index member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified element occurs too many times in the environment index member.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD007S **The XML structure of the *member_name* environment index member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times in the environment index member.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD008S **The XML structure of the *member_name* environment index member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified attribute occurs too many times in the environment index member.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD009S **The XML structure of the *member_name* environment index member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified attribute does not occur enough times in the environment index member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD010S **The XML structure of the *member_name* environment index member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:

Content was found in an attribute that cannot contain content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD011S **The XML structure of the *member_name* environment index member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

An attribute does not contain required content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD012S **The XML structure of the *member_name* environment index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:

An element contains too many characters. The name of the element and the maximum number of allowed characters are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD013S **The XML structure of the *member_name* environment index member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The environment index member contains an unknown attribute. The name of the unknown attribute and the name of the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD050S **The following LPAR serial number is duplicated in the environment index member: *serial_number*.**

Explanation:

The environment index member contains duplicate LPAR serial numbers. The duplicate serial number is indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD051S **The following DB2 serial number is duplicated in the environment index member: *serial_number*.**

Explanation:

The environment index member contains duplicate Db2 serial numbers. The duplicate serial number is indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD052S **The following Db2 group attach name is duplicated in the environment index member: *group_attach_name*.**

Explanation:

The environment index member contains duplicate group attach names.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD053S **The reference to the following DB2 subsystem for a Db2 group attach name is duplicated in the environment index member: *subsystem_ID*.**

Explanation:

The environment index member contains duplicate references to a Db2 subsystem for a Db2 group attach name.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD054S **The reference to the following DB2 subsystem for the *LPAR_name* LPAR is duplicated in the environment index member: *subsystem_ID*.**

Explanation:

The environment index member contains duplicate references to a Db2 subsystem for an LPAR. The duplicate subsystem ID is indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD055S **The following Db2 group attach name was not found in the environment index member: *group_attach_name*.**

Explanation:

A group attach name that is referenced by a Db2 member does not exist in the environment index member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD056S **The following LPAR was not found in the environment index member: *LPAR_name*.**

Explanation:

The LPAR does not exist in the environment index member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD057S **The following LPAR is duplicated in the environment index member: *LPAR_name*.**

Explanation:

The environment index member contains duplicate LPARs. The name of the duplicate LPAR name is indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD100W **The *member_name* product index member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the product index member is valid, the PL/I XML parser issued the specified exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception warning code.

CCQD101S **The *member_name* product index member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the product index member is valid, the PL/I XML parser issued the specified exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception error code. Ensure that the Tools Customizer data store data set DCB is the same as the sample SCCQSAMP(CCQCDATS) data set DCB.

CCQD102S **The XML structure of the *member_name* product index member is not valid. The *element_name* element is unknown.**

Explanation:

The specified product index member contains an unknown element.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD103S **The XML structure of the *member_name* product index member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

Content was found for an element that cannot contain content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD104S **The XML structure of the *member_name* product index member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element does not contain required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD105S **The XML structure of the *member_name* product index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD106S **The XML structure of the *member_name* product index member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified element occurs too many times in the product index member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD107S **The XML structure of the *member_name* product index member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times in the product index member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD108S **The XML structure of the *member_name* product index member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

An attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD109S **The XML structure of the *member_name* product index member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute does not occur enough times in the product index member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD110S **The XML structure of the *member_name* product index member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:
An attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD111S **The XML structure of the *member_name* product index member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:
An attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD112S **The XML structure of the *member_name* product index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD113S **The XML structure of the *member_name* product index member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:
The specified attribute in the product index member is unknown.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD118S **The content of the *member_name* product index member is not valid. The *configuration_ID* configuration ID for the *configuration-name* configuration name is not unique.**

Explanation:

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD120S **The content of the *member_name* product index member is not valid. The pack ID *pack_ID* that is referenced by product prefix *product_prefix* in the metadata library *library_name* could not be found.**

Explanation:
The specified pack ID could not be found in the metadata library.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQD121I **The specified pack contains the *component_name*, which was**

previously specified as a stand-alone product.

Explanation:

The specified component of the pack was previously specified as a stand-alone product.

System action:

None.

User response:

No action is required.

CCQD122I **The specified component metadata library was previously specified as part of the *pack_name*.**

Explanation:

The specified metadata library for the component was previously specified as part of a pack.

System action:

None.

User response:

No action is required.

CCQD123E **The customization library name *library_name* is being used by another product or component. Specify another customization library qualifier on the Tools Customizer Settings panel.**

Explanation:

A different product or component is using the specified customization library.

System action:

None.

User response:

Specify another customization library qualifier on the **Tools Customizer Settings** panel.

CCQD124E **The customization library *library_name* is in use by another metadata library.**

Explanation:

A different product or component is using the specified customization library. Specify another metadata library in the **Workplace** panel.

System action:

None.

User response:

Specify another customization library qualifier in the **Tools Customizer Settings** panel.

CCQD300W **The *member_name* product environment member is not valid. The PL/I XML parser issued the**

following exception warning code: *code_number*.

Explanation:

While determining if the product environment member is valid, the PL/I XML parser issued the specified exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception warning code.

CCQD301S **The *member_name* product environment member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the product environment member is valid, the PL/I XML parser issued the specified exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception error code.

CCQD302S **The XML structure of the *member_name* product environment member is not valid. The *element_name* element is unknown.**

Explanation:

The specified product environment member contains an unknown element.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD303S **The XML structure of the *member_name* product environment member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

Content was found for an element that cannot contain content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD304S **The XML structure of the *member_name* product environment member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element does not contain required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD305S **The XML structure of the *member_name* product environment member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD306S **The XML structure of the *member_name* product environment member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified element occurs too many times in the product environment member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD307S **The XML structure of the *member_name* product environment member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times in the product environment member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD308S **The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD309S **The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified attribute does not occur enough times in the product environment member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD310S **The XML structure of the *member_name* product environment member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:

The specified attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQD311S **The XML structure of the *member_name* product environment member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQD312S **The XML structure of the *member_name* product environment member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQD313S **The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute in the product environment member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQD350I **The *subsystem_ID* Db2 subsystem is associated with this product.**

Explanation:

The specified Db2 subsystem was added and saved in the Tools Customizer data store for the product to be customized.

System action:

Processing continues.

User response:

No action is required.

CCQD351I **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name is associated with this product.**

Explanation:

The specified Db2 member for the group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action:

Processing continues.

User response:

No action is required.

CCQD352I **The *group_attach_name* Db2 group attach name is associated with this product.**

Explanation:

The specified Db2 group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action:

Processing continues.

User response:

No action is required.

CCQD353E **The *subsystem_ID* Db2 subsystem is already associated with this product.**

Explanation:

The specified Db2 subsystem cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action:

None.

User response:

Ensure that the Db2 subsystem is specified correctly. If the problem persists, contact IBM Software Support.

CCQD354E **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name is already associated with this product.**

Explanation:

The specified Db2 member for the group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action:

None.

User response:

Ensure that the Db2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

CCQD355E **The *group_attach_name* Db2 group attach name is already associated with this product.**

Explanation:

The specified Db2 group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action:

Processing stops.

User response:

Ensure that the Db2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

CCQD356S **The *library_name* metadata library is already associated with the maximum number of allowed Db2 entries for this product.**

Explanation:

The specified metadata library cannot be associated with more Db2 entries because it is already associated with the number of Db2 entries that are allowed.

System action:

Processing stops.

User response:

Delete an associated Db2 entry, and associate the specified library with another Db2 entry again.

CCQD357I **The *subsystem_ID* Db2 subsystem is unassociated with this product.**

Explanation:

The specified Db2 SSID was unassociated with the product that you are customizing.

System action:

Processing continues.

User response:

No action is required.

CCQD358I **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name is unassociated with this product.**

Explanation:

The specified Db2 member for the Db2 group attach name was unassociated with the product that you are customizing.

System action:

Processing continues.

User response:

No action is required.

CCQD359I **The *group_attach_name* Db2 group attach name is unassociated with this product.**

Explanation:

The specified Db2 group attach name was unassociated with the product that you are customizing.

System action:

Processing continues.

User response:

No action is required.

CCQD360S **The *library_name* metadata library is not associated with the specified Db2 subsystem *subsystem_ID*.**

Explanation:

The specified Db2 subsystem and metadata library are not associated with each other.

System action:

None.

User response:

Ensure that the Db2 subsystem and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD361S **The *library_name* metadata library is not associated with the specified DB2 data sharing group member *member_name* for the *group_attach_name* Db2 group attach name.**

Explanation:

The specified Db2 data sharing group member for the group attach name and metadata library are not associated with each other.

System action:

None.

User response:

Ensure that the Db2 data sharing group member for the group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD362S **The *library_name* metadata library is not associated with the specified *group_attach_name* Db2 group attach name.**

Explanation:

The specified Db2 group attach name and metadata library are not associated with each other.

System action:

None.

User response:

Ensure that the Db2 group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD400W **The customization parser issued the *code_number* warning code while it parsed the product customization member *member_name*. See the PL/I programming guide for more information about this XML parser continuable exception code.**

Explanation:

While determining if the specified member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQD401S **The customization parser issued the *code_number* error code while it parsed the product customization member *member_name*. See the PL/I programming guide for more information about this XML parser terminating exception code.**

Explanation:

While determining if the specified member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQD500W **The *data_set_name* data store data set was not found.**

Explanation:

Tools Customizer could not find the specified data store data set.

System action:

None.

User response:

No action is required.

CCQD501W **The *data_set_name* data store data set was not found, so it was created.**

Explanation:

Tools Customizer created the specified data set because it could not be found.

System action:

None.

User response:

No action is required.

CCQD502E **The *data_set_name* data store data set is not writable.**

Explanation:

Tools Customizer cannot write to the specified data set.

System action:

None.

User response:

Ensure that the data set is writable.

CCQD503E **The *data_set_name* data store data set could not be opened with the *disposition_type* disposition.**

Explanation:

Tools Customizer could not open the data set with the specified disposition.

System action:

Processing stops.

User response:

Ensure that you have WRITE authority access to this data set.

CCQD504E **The *data_set_name* data store data set could not be opened with the *option_name* option.**

Explanation:

Tools Customizer could not open the data set with the specified option.

System action:

Processing stops.

User response:

Ensure that you have WRITE authority access to this data set.

CCQD505E **The data store data set *data_set_name* already exists in a different volume.**

Explanation:

Tools Customizer could not create the specified data set because the specified data set already exists in a different volume. Data store data set names must be unique.

System action:

Processing stops.

User response:

Specify a different data store data set name.

CCQD510I **The DB2 SSID and Db2 group attach name were created.**

Explanation:

The Db2 SSID and Db2 group attach name were created and saved in the data store.

System action:

None.

User response:

No action is required.

CCQD511E **The DB2 entry already exists in the list of Db2 entries to be associated.**

Explanation:

The Db2 entry cannot be added because it already exists in the list of Db2 entries to be associated.

System action:

None.

User response:

Specify a different Db2 entry.

CCQD512S **An error occurred while a DB2 entry was being created.**

Explanation:

A severe error occurred while a Db2 entry was being created.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD513E **The specified DB2 entry already exists and is associated with the current product on the Customizer Workplace panel.**

Explanation:

The Db2 entry cannot be added because it already exists, and it is already associated with the product to be customized.

System action:

None.

User response:

Press F3 to go to the **Customizer Workplace** panel to see the Db2 entry, or specify a different Db2 entry.

CCQD514E **A value is required for a DB2 subsystem, a Db2 group attach name, or both before they can be created.**

Explanation:

Required information is missing. A Db2 subsystem, a Db2 group attach name, or both must be specified.

System action:

None.

User response:

Specify a Db2 subsystem, a Db2 group attach name, or both.

CCQD515E **The specified DB2 entry already exists in the list of Db2 entries and is already associated with the current product.**

Explanation:

The Db2 entry has already been created and associated with the product that you want to customize.

System action:

None.

User response:

Specify a different Db2 entry.

CCQD516E **The specified DB2 entry already exists in the list of Db2 entries on the Associate DB2 Entry with Product panel but is not associated with the current product.**

Explanation:

The Db2 entry exists, but it must be associated with the product to be customized.

System action:

None.

User response:

On the **Customizer Workplace** panel, issue the ASSOCIATE command to associate the Db2 entry with the product.

CCQD517S **An error occurred while a DB2 entry was being copied.**

Explanation:

A severe error occurred while a Db2 entry was being copied

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQD518E **A value is required for a DB2 subsystem, a Db2 group attach name, or both before they can be copied.**

Explanation:

Required information is missing. A Db2 subsystem, a Db2 group attach name, or both must be specified.

System action:

None.

User response:

Specify a Db2 subsystem, a Db2 group attach name, or both.

CCQD519I **The DB2 entry was copied.**

Explanation:

The Db2 entry was copied and saved in the Tools Customizer data store.

System action:

None.

User response:

No action is required.

CCQD520S **The DB2 entry was copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.**

Explanation:

The Db2 entry was not completely copied because a product can be associated with only 1200 Db2 entries.

System action:

Processing stops.

User response:

Remove a Db2 entry from the list, and copy the specified Db2 entry again.

CCQD521E ***Line_command* is not a valid line command.**

Explanation:

The specified line command is not valid. Valid line commands are on the panel.

System action:

Processing stops.

User response:

Specify a valid line command.

CCQD522E **The *subsystem_ID* Db2 subsystem ID occurs more than once in the list. Each row must be unique.**

Explanation:

The specified Db2 subsystem ID can be used only once.

System action:

Processing stops.

User response:

Specify a different Db2 subsystem ID.

CCQD523E **The *group_attach_name* Db2 group attach name occurs more than once in the list. Each row must be unique.**

Explanation:

The specified Db2 group attach name can be used only once.

System action:

Processing stops.

User response:

Specify a different Db2 group attach name.

CCQD524E **The *member_name* Db2 member for the Db2 group attach name occurs more than once in the list. Each row must be unique.**

Explanation:

The specified Db2 member for the Db2 group attach name can be used only once.

System action:

Processing stops.

User response:

Specify a different Db2 member for the Db2 group attach name.

CCQD525I **The DB2 entries were created.**

User response:

No action is required.

CCQD526E **The *subsystem_ID* Db2 subsystem ID occurs more than once in the list. Each Db2 subsystem ID must be unique.**

Explanation:

The specified Db2 subsystem ID can be used only once.

System action:

Processing stops.

User response:

Specify a different Db2 subsystem ID.

CCQD527I **DB2 group attach names cannot be created during the copy process.**

Explanation:

The ability to create Db2 group attach names is not available during the copy process.

System action:

None.

User response:

Create Db2 group attach names by issuing the CREATE command on the **Customizer Workplace** panel.

CCQD528E **The *metadata_library* metadata library is already associated with *number* Db2 entries. The maximum number of associated Db2 entries for this metadata library is 256.**

Explanation:

A metadata library can be associated with a maximum of 256 Db2 entries. The specified metadata library is already associated with 256.

System action:

Processing stops.

User response:

Remove an existing association between the specified metadata library and a Db2 entry, and associate the specified the metadata library with another entry.

CCQD529I **At least one row is required.**

CCQD560E **The *subsystem_ID* Db2 subsystem already exists and is associated with the current product on the Customizer Workplace panel.**

Explanation:

The specified Db2 subsystem exists and is associated with the product that you are customizing.

System action:

None.

User response:

Specify another Db2 subsystem.

CCQD561E **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.**

Explanation:

The specified Db2 data sharing group for the Db2 group attach namer exists and is associated with the product that you are customizing.

System action:

None.

User response:

Specify another Db2 subsystem.

CCQD562E **The *group_attach_name* Db2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.**

Explanation:

The specified Db2 group attach name exists and is associated with the product that you are customizing. The subsystem is in the table on the **Customizer Workplace** panel.

System action:

None.

User response:

Specify another Db2 group attach name.

CCQD563E **A value is required for a DB2 subsystem, a Db2 group attach name, or both before they can be created.**

Explanation:

A Db2 subsystem, a Db2 group attach name, or both are not specified so one or both of them cannot be created.

System action:

None.

User response:

Specify a value for the Db2 subsystem, the Db2 group attach name, or both.

CCQD565E **The *subsystem_ID* Db2 subsystem already exists in the list of Db2 entries and is already associated with the current product.**

Explanation:

The specified subsystem is already associated.

System action:

None.

User response:

Specify a different Db2 subsystem.

CCQD566E **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name already exists in the list of Db2 entries and is already associated with the current product.**

Explanation:

The specified Db2member is already associated.

System action:

None.

User response:

Specify a different Db2 member.

CCQD567E **The *group_attach_name* Db2 group attach name already exists in the list of Db2 entries and is already associated with the current product.**

Explanation:

The specified Db2 group attach name is already associated.

System action:

None.

User response:

Specify another Db2 group attach name.

CCQD568I ***product_name* is not associated with a DB2 entry.**

Explanation:

The product that you are trying to customize is not associated with a Db2 entry. Before a product can be customized, it must be associated with at least one Db2 entry.

System action:

None.

User response:

Associate one or more Db2 entries with the product.

CCQD569I **The *product_name* product configuration is not associated with a DB2 entry.**

Explanation:

The configuration for the specified product is not associated with a Db2 entry.

System action:

None.

User response:

Associate one or more Db2 entries with the configuration.

CCQD577W **The *mode_name* Db2 mode of the *subsystem_ID* Db2 subsystem is not supported by the product.**

Explanation:

The product does not support the specified Db2 mode.

System action:

None.

User response:

Specify a supported Db2 mode.

CCQD578W **The *mode_name* Db2 mode of the *member_name* Db2 member for the Db2 group is not supported by the product.**

Explanation:

The product does not support the specified Db2 mode.

System action:

None.

User response:

Specify a supported Db2 mode.

CCQD579W **The *mode_name* Db2 mode of the *group_name* Db2 group attach name is not supported by the product.**

Explanation:

The product does not support the specified Db2 mode.

System action:

None.

User response:

Specify a supported Db2 mode.

CCQD580S **The *subsystem_ID* Db2 subsystem was copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.**

Explanation:

The copied Db2 subsystem was not associated with the product because the product is associated with the maximum number of Db2 entries.

System action:

None.

User response:

Remove an associated Db2 entry and associate the product with the copied Db2 subsystem.

CCQD581S **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name was copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.**

Explanation:

The copied Db2 member for the Db2 group attach name was not associated with the product because the product is associated with the maximum number of Db2 entries.

System action:

None.

User response:

Remove an associated Db2 entry and associate the product with the copied Db2 member.

CCQD582S **The *group_attach_name* Db2 group attach name was copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.**

Explanation:

The copied Db2 group attach name was not associated with the product because the product is associated with the maximum number of Db2 entries.

System action:

None.

User response:

Remove an associated Db2 entry and associate the product with the copied Db2 group attach name.

CCQD583I **The *from_DB2* Db2 subsystem was copied to the *to_DB2* subsystem.**

System action:

None.

User response:

No action is required.

CCQD584I **The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is copied to the *subsystem_ID* DB2 subsystem.**

Explanation:

The specified Db2 member was copied.

System action:

None.

User response:

No action is required.

CCQD585I **The *group_attach_name* Db2 group attach name cannot be copied because a Db2 member is required.**

Explanation:

The specified Db2 group attach name was not copied because a Db2 member was missing.

System action:

None.

User response:

No action is required.

CCQD586S **The current LPAR is *LPAR_name*, but the data store contains information about the *LPAR_name* LPAR. You must use the *LPAR_name* LPAR to customize the product.**

Explanation:

The LPAR that is stored in the data store data set must be used to customize the product.

System action:

Processing stops.

User response:

Use the LPAR that is stored in the data store data set.

CCQD587W **The *level_number* Db2 level of the *subsystem_name* Db2 subsystem is not supported by the product.**

Explanation:

The product does not support the specified Db2 level.

System action:

Processing continues.

User response:

Specify a supported level of Db2.

CCQD588W **The *level_number* Db2 level of the *member_name* Db2 member of the *group_name* Db2 group is not supported by the product.**

Explanation:

The product does not support the specified Db2 level.

System action:

Processing continues.

User response:

Specify a supported level of Db2.

CCQD589W **The *level_number* Db2 level of the *group_name* Db2 group attach name is not supported by the product.**

Explanation:

The product does not support the specified Db2 level.

System action:

Processing continues.

User response:

Specify a supported level of Db2.

CCQD593I **The *subsystem_ID* Db2 subsystem was deleted.**

User response:

No action is required.

CCQD594I **The *member_name* Db2 for the *group_attach_name* Db2 group attach name was deleted.**

User response:

No action is required.

CCQD595I **The *group_attach_name* Db2 group attach name was deleted.**

User response:

No action is required.

CCQD596E **The *subsystem_ID* Db2 subsystem was not deleted.**

Explanation:

An internal error occurred while the specified Db2 subsystem was being deleted.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083. Contact IBM Software Support.

CCQD597E **The *member_name* Db2 member for the *group_attach_name* Db2 group attach name was not deleted.**

Explanation:

An internal error occurred while the specified Db2 member was being deleted.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD598E **The *group_attach_name* Db2 group attach name was not deleted.**

Explanation:

An internal error occurred while the specified Db2 group attach name was being deleted.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD600W **The *member_name* product customization member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD601S **The *member_name* product customization member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD602S **The XML structure of the *member_name* product customization member is not valid. The *element_name* element is unknown.**

Explanation:

The data store member contains an unknown element.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD603S **The XML structure of the *member_name* product customization member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

The specified element cannot contain content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD604S **The XML structure of the *member_name* product customization member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element is missing required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD605S **The XML structure of the *member_name* product customization member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQD606S **The XML structure of the *member_name* product customization member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified element occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD607S **The XML structure of the *member_name* product customization member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified element does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD608S **The XML structure of the *member_name* product customization member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD609S **The XML structure of the *member_name* product customization member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD610S **The XML structure of the *member_name* product customization member is not valid. Content is not allowed for the *attribute_name* attribute in the**

***element_name* element, but content was found.**

Explanation:
The specified attribute cannot contain content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD611S **The XML structure of the *member_name* product customization member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:
The specified attribute does not contain required content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD612S **The XML structure of the *member_name* product customization member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.**

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD613S **The XML structure of the *member_name* product customization member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:
The specified attribute in the data store member is unknown.

System action:
Processing stops.

User response:

See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD614S **The content of the *member_name* product customization member is not valid. The value of the *element_name* element is not valid. The value is *value_name*.**

Explanation:

The specified value is not valid.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQD700W **The *member_name* Db2 data member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the XML structure of the Db2 data member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD701S **The *member_name* Db2 data member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the XML structure of the Db2 data member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD750W **The *value_number* value in the Db2 parameter *parameter_name* was skipped because only *maximum_number* values are allowed.**

Explanation:

The specified value was skipped because it exceeds the number of allowed values in the Db2 parameter.

System action:

Processing continues.

User response:

No action is required. To stop this message from being issued, remove the extra values from the Db2 parameter.

CCQD800W **The *member_name* LPAR data member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD801S **The *member_name* LPAR data member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD850W **The *value_number* value in the LPAR parameter *parameter_name* was skipped because only *maximum_number* values are allowed.**

Explanation:

The specified value was skipped because it exceeds the number of allowed values in the LPAR parameter.

System action:

Processing continues.

User response:

No action is required. To stop this message from being issued, remove the extra values from the LPAR parameter.

CCQD851I **The *subsystem_ID* Db2 subsystem is copied to the *member_name* Db2 member for the**

group_attach_name Db2 group attach name.

User response:

No action is required.

CCQD852I **The member_name Db2 member for the group_attach_name Db2 group attach name is copied to the member_name Db2 member for the group_attach_name Db2 group attach name.**

User response:

No action is required.

CCQD854I **The member_name Db2 member for the group_attach_name Db2 group 'attach name is copied to multiple Db2 entries.**

User response:

No action is required.

CCQD900W **The member_name product data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.**

Explanation:

While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD901S **The member_name product data member is not valid. The PL/I XML parser issued the following exception error code: code_number.**

Explanation:

While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD950W **The value_number value in the product parameter parameter_name was skipped because only maximum_number values are allowed.**

Explanation:

The specified value was skipped because it exceeds the number of allowed values in the product parameter.

System action:

Processing continues.

User response:

No action is required. To stop this message from being issued, remove the extra values from the product parameter.

CCQD960I **The subsystem_ID Db2 subsystem was changed to the member_name Db2 member for the group_attach_name Db2 group attach name.**

User response:

No action is required.

CCQD961I **The member_name Db2 member for the group_attach_name Db2 group attach name was changed to the subsystem_ID Db2 subsystem.**

User response:

No action is required.

CCQD962I **The member_name Db2 member for the group_attach_name Db2 group attach name was changed to the member_name Db2 member for the group_attach_name Db2 group attach name.**

User response:

No action is required.

CCQD963E **The DB2 group attach name cannot be blank when the Db2 subsystem ID is blank.**

Explanation:

A Db2 group attach name, Db2 subsystem ID, or both must be specified.

System action:

Processing stops.

User response:

Specify a Db2 group attach name, Db2 subsystem ID, or both.

CCQE000S **The specified message field name or message message_ID was not found.**

Explanation:

An error occurred while displaying a message field name or the specified message.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQE001E **An incorrect trace level was specified. Valid trace levels are 0 - 4.**

Explanation:

A wrong trace level was specified. Valid trace levels are 0 - 4.

System action:

Processing stops.

User response:

Specify a valid trace level 0 - 4.

CCQF028E **An asterisk was improperly specified in a filter argument.**

Explanation:

An asterisk, which is treated as data, is embedded in the filter arguments. A generic filter argument is specified by placing the asterisk in the last nonblank position of the argument. No rows match the filter arguments, so all rows will be shown.

System action:

Processing stops.

User response:

Specify a valid filter argument.

CCQF029I **More Db2 entries are associated with the specified product. All Db2 entries are listed.**

System action:

None.

User response:

No action is required.

CCQF080I **The customized jobs for the product that you are customizing are stored in this data set.**

System action:

None.

User response:

No action is required.

CCQF081I **The JCL must be browsed or edited.**

Explanation:

You can either browse or edit the JCL.

System action:

None.

User response:

No action is required.

CCQF082E **The *sort-command* command has an invalid sort field or order. The valid fields are *list-of-column-names*. The valid sort orders are A (for ascending) or D (for descending).**

Explanation:

An invalid sort field or order was specified.

System action:

Processing stops.

User response:

Specify a valid sort field or order.

CCQF083E **The *sort-command* command is missing a sort field.**

Explanation:

A sort field must be specified.

System action:

Processing stops.

User response:

Specify a valid sort field.

CCQF084E **The *sort-command* command has more than two sort fields specified.**

Explanation:

The specified sort command included more than two sort fields. The sort command can have up to two fields specified.

System action:

Processing stops.

User response:

Specify only one or two sort fields.

CCQF085E **A sort order was specified incorrectly in the *sort-command* command. A sort order can be specified only after a field name.**

Explanation:

Valid orders are A (for ascending) or D (for descending).

System action:

Processing stops.

User response:

Specify a valid sort order after a field name.

CCQF086E **The *sort-command* command has an invalid sort field. The valid fields are *list-of-the-table-column-names*.**

Explanation:

An invalid sort field was specified.

System action:
Processing stops.

User response:
Specify a valid sort field.

CCQF087E **The *sort-command* command has an invalid sort order. The valid orders are A (for ascending) or D (for descending).**

Explanation:
An invalid sort order was specified.

System action:
Processing stops.

User response:
Specify a valid sort order.

CCQF088E **No row match the specified filter argument. All rows are shown.**

Explanation:
No rows match the selected values.

System action:
Processing stops.

User response:
Specify a matched value for filtering.

CCQF089I **Type the search arguments to filter objects. A generic filter argument is a search argument of the form AA*.**

Explanation:
In a generic filter argument, only the characters up to the asterisk (*) are compared. The * must be placed in the last nonblank position of the argument. Asterisks embedded in the argument are treated as data.

System action:
None.

User response:
No action is required.

CCQF110I **To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.**

System action:
None.

User response:
No is action required.

CCQF111I **To show the Products to Customize section, specify a slash (/). To hide the Products to Customize section, remove the slash. The Product to Customize section can be shown**

or hidden only on the Customizer Workplace panel.

System action:
None.

User response:
No is action required.

CCQF112I **To show the Usage Notes section, specify a slash (/). To hide the Usage Notes section, remove the slash. The Usage Notes section can be shown only on the Product Parameters, LPAR Parameters, and DB2 Parameters panels.**

System action:
None.

User response:
No action is required.

CCQF113I **The specified values have been saved.**

System action:
None.

User response:
No action is required.

CCQF114I **Displays the Panel Display Options panel. Use this panel to select which information to display on panels .**

System action:
None.

User response:
No action is required.

CCQF116I **The fully qualified name of the data set into which you want to copy the current user profile. If the data set name exceeds 42 characters, enclose the name in quotation marks. ALTER or UPDATE authorization to this data set is required.**

System action:
None.

User response:
No action is required.

CCQF117I **The volume name in which the data set will reside. If left blank, the volume name will be decided by the system.**

System action:
None.

User response:
No action is required.

CCQF110I **To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.**

System action:
None.

User response:
No is action required.

CCQH001W **The specified option *option_name* is not valid.**

Explanation:
The option that was specified is not a valid option on the panel.

System action:
Tools Customizer stops.

User response:
Specify a valid option on the panel.

CCQH006W **Before you customize a product, verify your user settings.**

Explanation:
The user settings must be verified before a product can be customized.

System action:
Tools Customizer stops.

User response:
Verify the user settings.

CCQH007E **Check the user settings. One or more current values are not valid.**

Explanation:
One or more of the values in the user settings is not valid.

System action:
Tools Customizer stops.

User response:
Ensure that the specified values for the user settings are valid.

CCQH008W **Before you use Tools Customizer, you must select option 0 to verify your user settings.**

Explanation:
The user settings must be changed before a product can be customized.

System action:
Tools Customizer stops.

User response:
Change the user settings.

CCQH009E **You must select option 0 to change your user settings.**

Explanation:
User settings must be changed before a product can be customized.

System action:
Tools Customizer stops.

User response:
Change the user settings.

CCQI000W **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:
While determining if the Db2 parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:
Processing continues.

User response:
See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI001S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:
While determining if the Db2 parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action:
Processing stops.

User response:
See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI002S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *element name* element is unknown.**

Explanation:
The specified element in the Db2 parameter metadata member is unknown.

System action:
Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI003S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:
The specified element cannot contain content.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI004S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:
The specified element requires content.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI005S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI006S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.**

Explanation:

The specified element does not contain enough characters.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI007S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified element does not occur enough times.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI008S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI009S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute did not occur enough times.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI010S **The XML structure of the *member_name* Db2 parameter metadata member is not valid.**

Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation:

The specified attribute cannot have content.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI011S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute is missing required content.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI012S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI013S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute in the Db2 parameter metadata member is unknown.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI014S **The content of the *member_name* Db2 parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value of the element is not a valid value.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI015S **The content of the DB2 parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:

The specified value of the attribute is not a valid value.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI016S **The content of the DB2 parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type is not a valid data type.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI017S **The content of the DB2 parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:

The specified data type is not a valid data type.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI050S **The *member_name* Db2 parameter metadata member was not found in the *data_set_name* data set.**

Explanation:

Tools Customizer could not find the specified Db2 parameter metadata member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI051S **The *parameter_name* LPAR parameter in the *template_name* template does not have associated metadata in the *member_name* LPAR parameter metadata member.**

Explanation:

The specified template does not contain metadata for an LPAR parameter. The name of the LPAR parameter metadata member, the name of the LPAR parameter, and the name of the template are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI052S **The *parameter_name* product parameter in the *template_name* template does not have associated metadata in the *member_name* product parameter metadata member.**

Explanation:

The specified template does not contain metadata for a product parameter. The name of the product parameter metadata member, the name of the product parameter, and the name of the template are indicated in the message text.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI053E **The following metadata data set was not found: *data_set_name*.**

Explanation:

Tools Customizer could not find the specified metadata data set.

System action:

Processing stops.

User response:

Ensure that the metadata data set is specified correctly. If the problem persists, contact IBM Software Support.

CCQI054E **The following metadata data set could not be opened: *data_set_name*.**

Explanation:

Tools Customizer could not open the specified LPAR metadata data set.

System action:

Processing stops.

User response:

Ensure the metadata data set was specified correctly.

CCQI055S **The CCQ\$\$DB2 Db2 parameter metadata member was not found in the *data_set_name* Tools Customizer metadata data set.**

Explanation:

Tools Customizer could not find the Db2 parameter metadata member in the specified Tools Customizer metadata data set.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI056S **The CCQ\$\$LPR LPAR parameter metadata member was not found in the *data_set_name* data set.**

Explanation:

Tools Customizer could not find the specified LPAR parameter metadata member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI057S **The *member_name* product parameter metadata member was not found in the *data_set_name* data set.**

Explanation:

The product parameter metadata member was not found in the specified data set.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI058I *Product_name does not have any Db2 parameters.*

Explanation:

Db2 parameters are not required to customize the specified product.

System action:

Processing continues.

User response:

No action is required.

CCQI059I *Product_name does not have any LPAR parameters.*

Explanation:

LPAR parameters are not required to customize the specified product.

System action:

Processing continues.

User response:

No action is required.

CCQI060S *The parameter_name Db2 parameter in the task_description task condition does not have associated metadata in the member_name Db2 parameter metadata member.*

Explanation:

Associated metadata is missing for the specified Db2 parameter in a task.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI061S *The parameter_name LPAR parameter in the task_description task condition does not have associated metadata in the member_name LPAR parameter metadata member.*

Explanation:

Associated metadata is missing for the specified LPAR parameter in a task.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI062S *The parameter_name product parameter in the task_description task condition does not have associated metadata in the member_name product parameter metadata member.*

Explanation:

Associated metadata is missing for the specified product parameter in a task.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI063S *The parameter_name Db2 parameter in the task_description task and the step_description step does not have associated metadata in the member_name Db2 parameter metadata member.*

Explanation:

Associated metadata is missing for the specified Db2 parameter in a task and step.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI064S *The parameter_name LPAR parameter in the task_description task and the step_description step does not have associated metadata in the member_name LPAR parameter metadata member.*

Explanation:

Associated metadata is missing for the specified LPAR parameter in a task and step.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI065S *The parameter_name product parameter in the task_description task and the step_description step*

does not have associated metadata in the *member_name* parameter metadata member.

Explanation:

Associated metadata is missing for the specified parameter in a task and step.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI066S **The *parameter_name* Db2 parameter in the *task_description* task, *step_description* step, and *template_name* template condition does not have associated metadata in the *member_name* Db2 parameter metadata member.**

Explanation:

Associated metadata is missing for the specified Db2 parameter in a task, step, and template.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI067S **The *parameter_name* LPAR parameter in the *task_description* task, *step_description* step, and *template_name* template condition does not have associated metadata in the *member_name* LPAR parameter metadata member.**

Explanation:

Associated metadata is missing for the specified LPAR parameter in a task, step, and template.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI068S **The *parameter_name* product parameter in the *task_description* task, *step_description* step, and *template_name* template condition does not have associated metadata in the *member_name* product parameter metadata member.**

Explanation:

Associated metadata is missing for the specified product parameter in a task, step, and template.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI069S **Product metadata does not support multiple configurations, but the *template_name* product template contains the *parameter_name* parameter. Enable multiple configurations support for this product, and try again.**

Explanation:

The specified template contains a parameter for multiple configurations, but the product is not enabled to support multiple configurations.

System action:

Processing stops.

User response:

Enable multiple configurations support, and try again.

CCQI070E **The *parameter_name* Db2 parameter metadata member is not valid. The default length for the *parameter-element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.**

Explanation:

The specified length cannot be shorter than the default length.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI071E **The *parameter_name* LPAR parameter metadata member is not valid. The default length for the *parameter-element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default**

length will be truncated accordingly.

Explanation:

The specified length cannot be shorter than the default length.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI072E **The *parameter_name* product parameter metadata member is not valid. The default length for the *parameter-element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.**

Explanation:

The specified length cannot be shorter than the default length.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI073S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.**

Explanation:

The specified value already exists for an attribute.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI074S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.**

Explanation:

The specified value already exists for an attribute.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI075S **The XML structure of the *member_name* product parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.**

Explanation:

The specified value already exists for an attribute.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI076S **The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the Db2 parameter metadata member.**

Explanation:

The specified value already exists for an attribute.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI077S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the LPAR parameter metadata member.**

Explanation:

The specified parameter refers to a section that is not in the LPAR parameter metadata member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI078S **The XML structure of the *member_name* product parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the product parameter metadata member.**

Explanation:

The specified parameter refers to a section that is not in the product parameter metadata member.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI080S **The content of the *member_name* Db2 parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:

The specified value for an attribute in the Db2 parameter metadata member is not valid.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI081S **The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:

The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI082S **The content of the *member_name* product parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name***

element is incorrect. The value of the attribute is *value_name*.

Explanation:

The specified value for an attribute in the product parameter metadata member is not valid.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI088I **The *command* command is not active in BROWSE mode.**

Explanation:

The specified command can be entered only in Edit mode.

System action:

Processing continues.

User response:

Put the panel in Edit mode and reissue the command.

CCQI089I **The *command* command is already active.**

System action:

Processing continues.

User response:

No action required.

CCQI090S **The product-defined Db2 parameter *parameter_name* in the *member_name* parameter metadata member references the *section_ID* section ID, but this ID does not exist in either the parameter metadata member or the Db2 parameter metadata member.**

Explanation:

A section that does not exist in the parameter metadata member or the Db2 parameter metadata member is referenced by the specified Db2 parameter.

System action:

Processing stops.

User response:

See “Gathering diagnostic information” on page 1083. Contact IBM Software Support.

CCQI091S **The product-defined LPAR parameter in the *member_name* parameter metadata member references the *section_ID* section ID, but this ID does not exist in either the parameter metadata**

member or the LPAR parameter metadata member.

Explanation:

A section that does not exist in the parameter metadata member or the LPAR parameter metadata member is being referenced by the specified LPAR parameter.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI092S **The overridden DB2 parameter *parameter_name* in the *member_name* parameter metadata member does not exist in the Db2 parameter metadata member.**

Explanation:

The specified parameter does not exist.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI093S **The overridden LPAR parameter *parameter_name* in the *member_name* parameter metadata member does not exist in the LPAR parameter metadata member.**

Explanation:

The specified parameter does not exist.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI094S **The CCQ\$\$PRD product customization parameter metadata member was not found in the *data_set_name* data set.**

Explanation:

The specified data set must contain the CCQ\$\$PRD product customization parameter metadata member

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI100W **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI101S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI102S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *element_name* element is unknown.**

Explanation:

The specified element in the LPAR parameter metadata member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI103S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

The specified element cannot contain content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI104S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:
The specified element requires content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI105S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI106S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.**

Explanation:
The specified element does not contain enough characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI107S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The**

***element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified element does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI108S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI109S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute did not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI110S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:
The specified attribute cannot have content.

System action:
Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI111S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute is missing required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI112S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI113S **The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute in the LPAR parameter metadata member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI114S **The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an element in the LPAR parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI115S **The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:

The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI116S **The content of the *member_name* LPAR parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an element in the LPAR parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI117S **The content of the *member_name* LPAR parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an attribute in the LPAR parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI120S The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI121S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI122S The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI123S The XML structure of the *member_name* discover metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the

element_name element. The duplicate value is *value_name*.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI124S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI200W The XML structure of the *member_name* information metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation:
While determining if the information metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:
Processing continues.

User response:
See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI201S The XML structure of the *member_name* information metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation:
While determining if the information metadata member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI202S **The XML structure of the *member_name* information metadata member is not valid. The *element_name* element is unknown.**

Explanation:

The specified element in the information metadata member is unknown.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI203S **The XML structure of the *member_name* information metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

The specified element cannot contain content.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI204S **The XML structure of the *member_name* information metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element requires content.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI205S **The XML structure of the *member_name* information metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI206S **The XML structure of the *member_name* information metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.**

Explanation:

The specified element does not contain enough characters.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI207S **The XML structure of the *member_name* information metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI208S **The XML structure of the *member_name* information metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified attribute occurs too many times.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI209S **The XML structure of the *member_name* information**

metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation:

The specified attribute did not occur enough times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI210S **The XML structure of the *member_name* information metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:

The specified attribute cannot have content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI211S **The XML structure of the *member_name* information metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute is missing required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI212S **The XML structure of the *member_name* information metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI213S **The XML structure of the *member_name* information metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute in the information metadata member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI214S **The content of the *member_name* information metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an element in the information metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI215S **The content of the *member_name* information metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an attribute in the information metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI216S **The content of the *member_name* information metadata member is not valid because the data type of the *element_name* element is**

incorrect. The value is *value_name*.

Explanation:

The specified data type value for an element in the information metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI217S **The content of the *member_name* information metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an attribute in the information metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI218S **The content of the *member_name* information metadata member is not valid. The length of the *value_name* value that of the *attribute_name* attribute is longer than the *value_name* value of the *attribute_name* attribute.**

Explanation:

The first specified value cannot be longer than the second specified value.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI219S **The content of the *member_name* information metadata member is not valid. The *value_name* value of the *attribute_name* attribute contains the *value_name* value.**

Explanation:

The first specified value cannot be longer than the second specified value.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI220S **The XML structure of the *member_name* information metadata member is not valid. Content for the *attribute_name* attribute in the *element_name* element exceed *maximum_number* characters.**

Explanation:

The specified attribute contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI223S **The XML structure of the *member_name* information metadata member is not valid. The value that is specified for the Db2 Level already exists. The value is *value_name*.**

Explanation:

The specified value already exists.

System action:

Processing stops.

User response:

Specify a different Db2 level. If the problem persists, contact IBM Software Support.

CCQI224S **The XML structure of the *member_name* information metadata member is not valid. The value that is specified for the Db2 Mode already exists. The value is *value_name*.**

Explanation:

The specified value already exists.

System action:

Processing stops.

User response:

Specify a different Db2 mode. If the problem persists, contact IBM Software Support.

CCQI250S **The information metadata member was not found in the *data_set_name* data set.**

Explanation:

Tools Customizer could not find the information metadata member in the specified data set.

System action:

Processing stops.

User response

If this message was issued on the **Specify the Metadata Library** (CCQPHLQ) panel, specify the product metadata library. The name of this library is *hlq.SCKZDENU*.

Do not specify the Tools Customizer metadata library, which is *hlq.SCCQDENU*.

If the problem persists, identify the name of the Tools Customizer trace data set and contact IBM Software Support.

CCQI251E **The *member_name* member was not accessible in the *data_set_name* data set.**

Explanation:

The specified member could not be accessed in the data set.

System action:

Processing stops.

User response:

Specify the correct metadata library.

CCQI252S **The information metadata member was not found in the *library_name* component metadata library that is part of the *library_name* pack metadata library. The name of the pack is *pack_name*.**

Explanation:

The specified component metadata library does not contain the information metadata member.

System action:

Processing stops.

User response:

Specify the correct metadata library.

CCQI253E **The *library_name* Tools Customizer metadata library is not current. Update the metadata library on the Tools Customizer Settings panel.**

Explanation:

The specified metadata library is not current.

System action:

Processing stops.

User response:

Specify a current metadata library on the **Tools Customizer Settings** panel.

CCQI300W **The XML structure of the *member_name* sequence metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI301S **The XML structure of the *member_name* sequence metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code, and contact IBM Software Support.

CCQI302S **The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element is unknown.**

Explanation:

The specified element in the sequence metadata member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083. Contact IBM Software Support.

CCQI303S **The XML structure of the *member_name* sequence metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

The specified element cannot contain content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI304S **The XML structure of the *member_name* sequence metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:
The specified element is missing required content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI305S **The XML structure of the *member_name* sequence metadata member is not valid. Content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI306S **The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified element occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI307S **The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI308S **The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI309S **The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI310S **The XML structure of the *member_name* sequence metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:
The specified attribute cannot contain content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI311S **The XML structure of the *member_name* sequence metadata member is not valid. Content is required for the *attribute_name***

attribute in the *element_name* element, but content was not found.

Explanation:

The specified attribute is missing required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI312S **The XML structure of the *member_name* sequence metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI313S **The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute in the sequence metadata member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI314S **The content of the *member_name* sequence metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an element in the sequence metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI315S **The content of the *member_name* sequence metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an attribute in the sequence metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI316S **The content of the *member_name* sequence metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an element in the sequence metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI317S **The content of the *member_name* sequence metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an attribute in the sequence metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI350S **The XML structure of the *member_name* sequence metadata member is not valid because the value of the *attribute_name* attribute in the *element_name***

element is incorrect. The value is *value_name*.

Explanation:

A specified value for an attribute in the sequence metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI351S **The *member_name* sequence metadata member was not found in the *data_set_name* metadata data set.**

Explanation:

Tools Customizer could not find the specified sequence metadata member in the metadata data set.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI352S **The *template_name* product template was not found in the *data_set_name* metadata data set.**

Explanation:

Tools Customizer could not find the specified product template in the data set.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI353S **The sequence metadata member was not found in the *data_set_name* component data set that is part of the *data_set_name* pack.**

Explanation:

Tools Customizer could not find the sequence metadata member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI360S **The XML structure of the *member_name* sequence metadata member is not valid. The value of the *attribute_name* attribute in the**

***element_name* element already exists.**

Explanation:

The specified attribute contains a value that already exists.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI361S **The XML structure of the *member_name* sequence metadata member is not valid. The condition element on the *level_type* level already contains a relational operator.**

Explanation:

A relational operator already exists for the condition element on the specified level.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI362S **The XML structure of the *member_name* sequence metadata member is not valid. The condition element on the *level_type* level must contain only one content string or content number element.**

Explanation:

Only one content string element or content number element can be contained in the condition element on the specified level.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI363S **The XML structure of the *member_name* sequence metadata member is not valid. The condition element in the *element_name* element with the *attribute_name* attribute must contain either the content string element or the content number element.**

Explanation:

Either the content string element or the content number element must be in the condition element.

System action:

Processing stops.

User response:

Contact IBM Software Support.

CCQI400W **The XML structure of the *member_name* parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining the parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI401S **The XML structure of the *member_name* parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI402S **The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element is unknown.**

Explanation:

The specified element in the parameter metadata member is unknown.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI403S **The XML structure of the *member_name* parameter metadata member is not valid. Content is not allowed for the**

***element_name* element, but content was found.**

Explanation:

The specified element cannot contain content.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI404S **The XML structure of the *member_name* parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element requires content.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI405S **The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI406S **The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.**

Explanation:

The specified element does not contain enough characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI407S **The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI408S **The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified attribute occurs too many times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI409S **The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified attribute does not occur enough times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI410S **The XML structure of the *member_name* parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:

The specified attribute cannot have content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI411S **The XML structure of the *member_name* parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute is missing required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI412S **The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI413S **The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute in the parameter metadata member is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI414S **The content of the *member_name* parameter metadata member is not valid because the value of the *element_name* element is**

incorrect. The value is *value_name*.

Explanation:

The specified value for an element in the parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI415S **The content of the *member_name* parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an attribute in the parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI416S **The content of the *member_name* parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an element in the parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI417S **The content of the *member_name* parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an attribute in the parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI420S **The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element is unknown for the overridden Db2 parameter.**

Explanation:

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI421S **The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element is unknown for the overridden LPAR parameter.**

Explanation:

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI422S **The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown for the overridden Db2 parameter.**

Explanation:

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI423S **The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown for the overridden LPAR parameter.**

Explanation:

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI450S **The *member_name* product parameter metadata member was not found in the *data_set_name* data set.**

Explanation:

Tools Customizer could not find the specified product parameter metadata member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI510W **The *data_set_name* data store data set does not exist.**

Explanation:

The specified data store data set does not exist.

System action:

Processing continues.

User response:

Ensure that the data store data set exists.

CCQI511S **The *data_set_name* data store data set cannot be opened by using the *disposition_type* disposition.**

Explanation:

The specified data store data set could not be opened with the specified disposition.

System action:

Processing continues.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI512S **The *data_set_name* data store data set cannot be opened by using the *option-type* option.**

Explanation:

The specified data store data set was unable to be opened with the specified option.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI600W **The XML structure of the *member_name* product customization parameter**

metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation:

While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI601S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:

While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI602S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element is unknown.**

Explanation:

The specified product customization parameter metadata member contains an unknown element.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI603S **The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:

Content was found in an element that cannot contain content.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI604S **The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:
The specified element does not contain required content.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI605S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI606S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified element occurs too many times in the product customization parameter metadata member.

System action:
Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI607S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified element does not occur enough times in the product customization parameter metadata member.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI608S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:
The specified attribute occurs too many times in the product customization parameter metadata member.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI609S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute does not occur enough times in the product customization parameter metadata member.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI610S **The XML structure of the *member_name* product**

customization parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation:

Content was found in an element that cannot contain content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI611S **The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute does not contain required content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI612S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The content length for the *attribute_name* attribute in the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified attribute contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI613S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the**

***element_name* element is unknown.**

Explanation:

The specified product customization parameter metadata member contains an unknown attribute.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI614S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The value of the *element_name* element is not valid. The value *value_name*.**

Explanation:

The specified value of the element is not a valid value.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI615S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The value of the *attribute_name* attribute for the *element_name* element is not valid. The value is *value_name*.**

Explanation:

The specified value of the attribute is not a valid value.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQI616S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The data type of the *element_name* element is 'not valid. The value of the element is *value_name*.**

Explanation:

The specified data type is not a valid data type.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI617S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The data type of the *attribute_name* attribute for the *element_name* element is not valid. The value of the attribute is *value_name*..**

Explanation:

The specified data type is not a valid data type.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI650S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.**

Explanation:

The specified value for an attribute already exists.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI651S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *parameter_name* parameter refers to the following section, which was not found in the *member_name* product customization parameter metadata member: *section-name*.**

Explanation:

The specified section is not in the product customization parameter metadata member.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI652S **The *member_name* product customization metadata member not valid. The default length for the *element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.**

Explanation:

The specified length cannot be shorter than the default length.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI653S **The content of the *member_name* product customization parameter metadata member is not valid. The value of the *attribute_name* attribute in the *element_name* element is not valid. The value of the attribute is *value_name*.**

Explanation:

The specified value of the attribute is not a valid value.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information”](#) on page 1083.
Contact IBM Software Support.

CCQI700W **The XML structure of the *member_name* solution pack metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:

While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:

Processing continues.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI701S **The XML structure of the *member_name* solution pack metadata member is not valid. The PL/I XML parser issued the**

following exception error code:
code_number.

Explanation:

While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception error code.

System action:

Processing stops.

User response:

See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQI702S **The XML structure of the *member_name* solution pack metadata member is not valid. The *element_name* element is unknown.**

Explanation:

The specified solution pack metadata member contains an unknown element.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI703S **The XML structure of the *member_name* solution pack metadata member is not valid. Content is not allowed for the *element_name* element, but content was found**

Explanation:

Content was found in an element that cannot contain content.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI704S **The XML structure of the *member_name* solution pack metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:

The specified element does not contain required content.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI705S **The XML structure of the *member_name* solution pack metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.**

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI706S **The XML structure of the *member_name* solution pack metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified element occurs too many times.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI707S **The XML structure of the *member_name* solution pack metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times.

System action:

Processing stops.

User response:

See “[Gathering diagnostic information](#)” on page 1083. Contact IBM Software Support.

CCQI708S **The XML structure of the *member_name* solution pack metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI709S **The XML structure of the *member_name* solution pack metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.**

Explanation:
The specified attribute does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI710S **The XML structure of the *member_name* solution pack metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:
The specified attribute cannot have content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI711S **The XML structure of the *member_name* solution pack metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:
The specified attribute is missing content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI712S **The XML structure of the *member_name* solution pack metadata member is not valid. The content length for the *attribute_name* attribute in the**

***element_name* element cannot exceed *maximum_number* characters.**

Explanation:
The specified attribute contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI713S **The XML structure of the *member_name* solution pack metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:
The specified attribute in the solution pack metadata member is unknown.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI714S **The XML structure of the *member_name* solution pack metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:
The specified value of the element is not a valid value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 1083.
Contact IBM Software Support.

CCQI715S **The XML structure of the *member_name* solution pack metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:
The specified value of the attribute is not a valid value.

System action:
Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI716S **The XML structure of the *member_name* solution pack metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type is not a valid data type.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI717S **The XML structure of the *member_name* solution pack metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.**

Explanation:

The specified data type is not a valid data type.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI720S **The XML structure of the *member_name* solution pack metadata member is not valid. The *msg* element is required for the *component_name* component that is not customizable.**

Explanation:

The *msg* element is required for the specified component, which cannot be customized by using Tools Customizer.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI750S **The solution pack metadata member was not found in the *library_name* metadata library.**

Explanation:

Tools Customizer could not find the solution pack metadata member in the specified library.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI751S **The version in the *library_name* solution pack metadata library is different than the version in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.**

Explanation:

The version in the solution pack metadata library does not match the version in the component metadata library.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI752S **The release in the *library_name* solution pack metadata library is different than the release in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.**

Explanation:

The release in the solution pack metadata library does not match the release in the component metadata library.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI753S **The modification level in the *library_name* solution pack metadata library is different than the modification level in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.**

Explanation:

The modification level in the solution pack metadata library does not match the modification level in the component metadata library.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQI755S **The XML structure of the *member_name* parameter metadata member is not valid. When a default value is not specified in the metadata member, the "required" attribute with a value of "true" cannot be specified.**

System action:
Processing stops.

User response:
Contact IBM Software Support.

CCQM002E **The *command_name* line command is not valid: .**

Explanation:
The specified line command is not valid.

System action:
Processing continues.

User response:
Specify a valid line command on the panel.

CCQ0000W **The XML structure of the *member_name* discover parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.**

Explanation:
While determining if the discover parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:
Processing continues.

User response:
See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQ0001S **The XML structure of the *member_name* discover parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.**

Explanation:
While determining if the Discover metadata member is valid, the PL/I XML parser issued an exception error code.

System action:
Processing stops.

User response:
See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.
Contact IBM Software Support.

CCQ0002S **The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element is unknown.**

Explanation:
The specified element in the discover parameter metadata member is unknown.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0003S **The XML structure of the *member_name* discover parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.**

Explanation:
The specified element cannot contain content.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0004S **The XML structure of the *member_name* discover parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.**

Explanation:
The specified element is missing required content.

System action:
Processing stops.

User response:
See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0005S **The XML structure of the *member_name* discover parameter metadata member is not valid. The content length for the *element_name* element cannot**

exceed *maximum_number* characters.

Explanation:

The specified element contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0006S **The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified element occurs too many times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0007S **The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.**

Explanation:

The specified element does not occur enough times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0008S **The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.**

Explanation:

The specified attribute occurs too many times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0009S **The XML structure of the *member_name* discover parameter**

metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation:

The specified attribute does not occur enough times.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0010S **The XML structure of the *member_name* discover parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.**

Explanation:

The specified attribute cannot contain content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0011S **The XML structure of the *member_name* discover parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.**

Explanation:

The specified attribute requires content.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0012S **The XML structure of the *member_name* discover parameter metadata member is not valid. The content length for the *attribute_name* attribute in the *element_name* element in the cannot exceed *maximum_number* characters.**

Explanation:

The specified attribute contains too many characters.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0013S **The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.**

Explanation:

The specified attribute is unknown.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0014S **The content of the *member_name* discover parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

A The specified value for an element in the discover parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0015S **The content of the *member_name* discover parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified value for an attribute in the discover parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0016S **The content of the *member_name* discover parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an element in the discover parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0017S **The content of the *member_name* product parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.**

Explanation:

The specified data type value for an attribute in the product parameter metadata member is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#).
Contact IBM Software Support.

CCQ0050S **The *data_set_name* Discover REXX EXEC data set could not be initialized or was not found.**

Explanation:

Tools Customizer could not find or could not initialize the specified Discover REXX EXEC data set.

System action:

Processing stops.

User response:

Ensure that the Discover REXX EXEC is specified correctly.

CCQ0051W **The *data_sharing_group_ID* data sharing group ID cannot contain more than four characters.**

Explanation:

The specified data sharing group ID contains too many characters.

System action:

Processing continues.

User response:

Ensure that the specified data sharing group ID does not exceed four characters.

CCQ0052S **The *REXX_EXEC_name* Discover REXX EXEC was not found in the *data_set_name* Discover data set.**

Explanation:

Tools Customizer could not find the Discover REXX EXEC in the specified data set.

System action:
Processing stops.

User response:
Ensure that the Discover data set was specified correctly.

CCQ0053W **The *LPAR_name* LPAR name cannot contain more than eight characters.**

Explanation:
The specified LPAR name contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified LPAR name does not exceed eight characters.

CCQ0054W **The *subsystem_ID* Db2 SSID cannot contain more than four characters. The record was not processed.**

Explanation:
The specified Db2 SSID contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified Db2 SSID does not exceed four characters.

CCQ0055W **The *parameter_name* Db2 group attach name parameter is in the *record_name* Discover record, but a Db2 group attach name was not specified. The record was not processed.**

Explanation:
The Discover record contains a data sharing group parameter, but a Db2 group attach name was not specified.

System action:
Processing continues.

User response:
Ensure that information is specified correctly on the **Discover Customized Product Information** panel.

CCQ0056W **The *parameter_name* Db2 parameter in the *record_name* Discover record did not have a Db2 group attach name or a Db2 SSID. The record was not processed.**

Explanation:

The Discover record did not have a Db2 group attach name or a Db2 subsystem ID in the Db2 parameter.

System action:
Processing continues.

User response:
Ensure that information is specified correctly on the **Discover Customized Product Information** panel.

CCQ0057W **The Discover EXEC could not find the *parameter_name* parameter in the metadata for the product to be customized. The record was not processed.**

Explanation:
The specified parameter could not be found in the metadata for the product to be customized.

System action:
Processing continues.

User response:
Ensure that information is specified correctly on the **Discover Customized Product Information** panel.

CCQ0058W **The *parameter_name* product parameter name in the *record_type* Discover record does not start with *CCQ_LPR_*, *CCQ_DB2_*, or *CCQ_PRD_*. The record was not processed.**

Explanation:
The parameter in the record does not start with *CCQ_DB2_*, *CCQ_LPAR_*, or *CCQ_PRD_*.

System action:
Processing continues.

User response:
See [“Gathering diagnostic information”](#) on page 1083. Contact IBM Software Support.

CCQ0059W **The *parameter_name* product parameter cannot contain more than 72 characters. The record was not processed.**

Explanation:
The specified product parameter contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified product parameter does not exceed 72 characters.

CCQ0060W **The *record_name* Discover record from the REXX EXEC output must start with the following record**

type: *record_type*. The record was not processed.

Explanation:

A Discover record from the REXX EXEC output must start with the specified Db2 record type.

System action:

Processing continues.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0061I **If you do not have a previously customized version of the product, do not run the Discover EXEC. Press END to go to the Customizer Workplace panel.**

Explanation:

This message is issued when you customize a product for a the first time. It prompts you to use the Discover EXEC to discover data from a previous customization of the specified product.

System action:

Processing continues.

User response

Tip: Using the Discover EXEC saves time and reduces errors that can error when parameters are specified manually. If you want to use the Discover EXEC, specify the required information on the **Discover Customized Product Information** panel. Otherwise, press End to continue without discovering data from a previous customization of the product.

CCQ0062W **The Discover EXEC could not find the following *parameter_name* parameter in the Db2 metadata. The record was not processed.**

Explanation:

The specified parameter is missing in the Db2 metadata.

System action:

Processing continues.

User response:

If this parameter is required, contact IBM Software Support.

CCQ0064W **The *Discover-record* Discover record did not have a parameter name. The record was not processed.**

Explanation:

A parameter name was missing in the Discover record.

System action:

Processing continues.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0065W **The value for the *parameter_name* parameter is ignored because it has more than *maximum_number* characters, which is the maximum length that is defined in the metadata. The value is *parameter_value*.**

Explanation:

The specified value exceeded the maximum allowed length, which was defined in the metadata. Tools Customizer truncated the extra characters.

System action:

Processing continues.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0066W **The *record_name* Discover record from the Discover REXX EXEC output does not have a parameter value. The record was not processed.**

Explanation:

The Discover record was missing a parameter value from the Discover EXEC output.

System action:

Processing continues.

User response:

Ensure that information was specified correctly on the **Discover Customized Product Information** panel.

CCQ0067W **The *parameter_name* parameter is defined in the metadata to support one value, but more than one value was found. The last value was used.**

Explanation:

The definition of the parameter in the metadata supports one value, but more than one value was specified. Only the last value was used.

System action:

Processing continues.

User response:

Ensure that information was specified correctly on the **Discover Customized Product Information** panel.

CCQ0068W **The value of the *parameter_name* parameter is ignored because the parameter is defined as**

internal=true. The value is *value_name*.

Explanation:

The specified value of the parameter is ignored because it is defined as internal=true.

System action:

Processing continues.

User response:

Ensure that information was specified correctly on the **Discover Customized Product Information** panel.

CCQ0069W **The Discover EXEC did not find the *parameter_name* parameter in the LPAR metadata. The record was not processed.**

Explanation:

The specified parameter is missing from the LPAR metadata.

System action:

Processing continues.

User response:

Ensure that information was specified correctly on the **Discover Customized Product Information** panel.

CCQ0070W **The *record_type* Discover record contains an incorrect delimiter between the Environment section and the Data section. The record was not processed.**

Explanation:

Tools Customizer found an incorrect delimiter between the Environment section and the Data section.

System action:

None.

User response:

No action is required.

CCQ0071W **The *member_name* member could not be found in the *data_set_name* Discover data set.**

Explanation:

Tools Customizer could not find the specified Discover data set.

System action:

None.

User response:

No action is required.

CCQ0072S **The *member_name* discover metadata member was not found in the *data_set_name* metadata data set.**

Explanation:

Tools Customizer could not find the specified metadata member in the data set.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0073E **The *member_name* discover metadata member is not valid because the default length for the *element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.**

Explanation:

The default length for the specified parameter element is longer than the parameter.

System action:

Processing continues.

User response:

No action is required.

CCQ0074S **The content of the *member_name* discover metadata member is not valid. The value of the *attribute_name* attribute in the *element_name* element is not valid. The value of the attribute is *value_name*.**

Explanation:

The specified value is not valid.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0075W **The *configuration_ID* configuration ID in the *record_name* Discover record is incorrect. The record was not processed.**

Explanation:

The specified configuration ID is not correct.

System action:

Processing continues.

User response:

No action is required.

CCQ0076W **The *configuration_ID* configuration ID cannot contain more than**

maximum_number characters. The record was not processed.

Explanation:

The specified configuration ID contains too many characters.

System action:

Processing continues.

User response:

No action is required.

CCQ0077S **The discover metadata member was not found in the *data_set_name* component data set that is part of the *data_set_name* pack.**

Explanation:

The discover metadata member was not found in the specified component data set.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQ0078I **Additional configurations were discovered and saved in the data store. All Db2 entries associated with this configuration are listed.**

System action:

None.

User response:

No action is required.

CCQ0080I ***Product_name* does not support the Discover process.**

Explanation:

The specified product does not support the Discover process.

System action:

None.

User response:

No action is required.

CCQP000E **The value of the *mode_name* Db2 mode is not valid for the *level_name* Db2 level.**

Explanation:

The specified Db2 mode is not valid for the Db2 level.

System action:

Processing stops.

User response:

Specify a valid Db2 mode for the Db2 level.

CCQP001E **The value of the *mode_name* Db2 mode is missing.**

Explanation:

The specified Db2 mode is not defined.

System action:

Processing stops.

User response:

Specify a value for the Db2 mode.

CCQP002E **The value of the *mode_name* Db2 level is missing.**

Explanation:

The specified Db2 level is not defined.

System action:

Processing stops.

User response:

Specify a value for the Db2 level.

CCQP003E **The value of the *level_name* Db2 level is not valid.**

Explanation:

The specified Db2 level does not have a valid name.

System action:

Processing stops.

User response:

Specify a valid value for the Db2 level.

CCQP004S **The *parameter_name* parameter does not exist in the CCQ\$\$DB2 Db2 parameter metadata member.**

Explanation:

The CCQ\$\$DB2 Db2 parameter metadata member does not contain the specified parameter.

System action:

Processing stops.

User response:

See [“Gathering diagnostic information” on page 1083](#). Contact IBM Software Support.

CCQP005E **The value of the *subsystem_ID* Db2 SSID is missing.**

Explanation:

The specified Db2 SSID is not defined.

System action:

Processing stops.

User response:

Specify a valid value for the Db2 SSID.

CCQP006E **The value of the *group_attach_name* Db2 group attach name is missing.**

Explanation:

The specified Db2 group attach name is not defined.

System action:

Processing stops.

User response:

Specify a valid Db2 group attach name.

CCQQ000E **Specify a valid metadata library. Each qualifier of the library must start with an alphabetic character and must be 1-8 alphanumeric characters. The library name must be 1-44 characters.**

Explanation:

The metadata library was not specified in the correct format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action:

Tools Customizer prompts for the correct library name.

User response

Specify a library in the correct format. If the message was issued on the **Specify the Metadata Library** (CCQPHLQ) panel, specify the product metadata library. The name of this library is *hlq.SCKZDENU*.

Do not specify the Tools Customizer metadata library, which is *hlq.SCCQDENU*.

CCQQ001E **The *data_set_name* data set name that was specified for the metadata library was not found.**

Explanation:

The data set does not exist, or the data set name was written in the incorrect format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action:

Tools Customizer prompts for the correct data set name.

User response:

Specify a data set name in the correct format.

CCQQ002E **The data set name that was specified for the *library_name* metadata library cannot be opened.**

Explanation:

Tools Customizer could not open the data set.

System action:

Tools Customizer prompts for an available data set.

User response:

Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ003E **The *data_set_name* data set name that was specified for the metadata sample library is not valid. The data set must be in the following format: HLQ.SxxxSAMP.**

Explanation:

The specified data set name was not specified in the correct format.

System action:

None.

User response:

Specify the data set name in the following format: HLQ.SxxxSAMP, where xxx is the three-character prefix for the product.

CCQQ004E **The *data_set_name* data set is being used by another user. Try again when the data set is not being used.**

Explanation:

Another user is using the specified data set.

System action:

None.

User response:

Ensure that the specified data set is not being used.

CCQQ009E **The *data_set_name* data set name that was specified for the metadata library is not valid because the data set is empty.**

Explanation:

The specified data set is empty.

System action:

Tools Customizer prompts for an available data set.

User response:

Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ011E **The *library_name* metadata library for the component that is part of the *library_name* pack was not found in the catalog. The name of the pack is *pack_name*, and the name of the component is *component_name*.**

Explanation:

The specified metadata library is not in the catalog.

System action:

None.

User response:

Specify another metadata library.

CCQQ012E **The *library_name* metadata library for the component that is part of the *library_name* pack cannot be opened.**

Explanation:

The specified metadata library cannot be opened.

System action:

None.

User response:

Ensure that the name of the library is specified correctly.

CCQS000I **Tools Customizer is being invoked for the first time or the previous ISPF session ended before Tools Customizer was exited. In both cases, the fields on this panel are populated with default values. Review these default values or specify new values to be used to customize products or packs.**

Explanation:

When you customize a stand-alone product or a solution pack for the first time, or when an ISPF session unexpectedly ends before the ISPF profile is saved, you must specify or review your Tools Customizer user settings.

System action:

Processing stops.

User response:

Review and accept the default settings, or specify new settings.

CCQS001E **The following command is not valid: *command_name*.**

Explanation:

The specified command is not a valid command on the panel.

System action:

Processing stops.

User response:

Specify a valid command.

CCQS002W **The *data_set_name* Discover data set could not be found.**

Explanation:

Tools Customizer could not find the specified data set.

System action:

The data set will be allocated, and processing continues.

User response:

Ensure that the data set name is specified correctly because the data set will be allocated with this name after the values are saved.

CCQS003W **The *data_set_name* Discover data set was not found so it was created.**

Explanation:

Tools Customizer could not find the specified data set.

System action:

Processing continues.

User response:

Ensure that the data set name is specified correctly.

CCQS004I **The settings were saved.**

Explanation:

The settings that you changed were saved.

System action:

Processing continues.

User response:

No action is required.

CCQS006W **The length of a qualifier for the *data_set_name* customization library data set exceeds 26 characters.**

Explanation:

The qualifier for the customization library data set is too long. The qualifier cannot exceed 26 characters.

System action:

Processing continues.

User response:

Specify a qualifier that is 26 characters or less.

CCQS007E **The discover data set *data_set_name* could not be opened with the *option-type* option.**

Explanation:

The specified option could not open the Discover data set.

System action:

None.

User response:

Specify a data set to which you have WRITE access.

CCQS008E **The Discover data set *data_set_name* exists on a different volume.**

Explanation:

The specified Discover data set must exist on the same volume as where it was created.

System action:

Processing continues.

User response:

Specify a different Discover data set name.

CCQS010E The customization library qualifier is not valid.

Explanation:

The customization library qualifier that was specified is not valid.

System action:

None.

User response:

Specify a valid qualifier for the customization library.

CCQS011E The group attach option is not valid.

Explanation:

The group attach option that was specified is not valid.

System action:

None.

User response:

Specify a valid option for the group attach option.

CCQS012E The Tools Customizer metadata library is not valid.

Explanation:

The metadata library that was specified is not a valid data set.

System action:

None.

User response:

Specify a valid data set for the metadata library.

CCQS013E The Discover data set is not valid.

Explanation:

The Discover data set that was specified is not a valid data set.

System action:

None.

User response:

Specify a valid Discover data set.

CCQS014E The data store data set is not valid.

Explanation:

The data set that was specified is not a valid data set.

System action:

None.

User response:

Specify a valid data store data set.

CCQS015E Tools Customizer is already running.

Explanation:

A session of Tools Customizer is already running in your environment. Only one Tools Customizer session is allowed.

System action:

None.

User response:

The trace data set is being used. Free the trace data set, and start Tools Customizer again.

CCQS018E Information on the first line of the job card exceeds 57 characters.

Explanation:

The first line of the job card can contain only 57 characters. This character limit includes a continuation character.

System action:

Tools Customizer clears the first line of the job card.

User response:

Specify information that does not exceed 57 characters on the first line of the job card.

CCQS019E The required trace data set, *data_set_name*, is currently not accessible.

Explanation:

The trace data set must be accessible.

System action:

Processing stops.

User response:

Ensure that the trace data set is accessible.

CCQS020E An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

Explanation:

To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action:

None.

User response:

Ensure that ALTER authority for the specified customization library data set is granted.

CCQS021E The value *value_name* in the field that contains the cursor position is not valid.

Explanation:

The specified value is not valid.

System action:

None.

User response:

Specify a valid value.

CCQS022E **An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.**

Explanation:

To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action:

None.

User response:

Ensure that UPDATE authority for the specified customization library data set is granted.

CCQS023E **An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.**

Explanation:

To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action:

None.

User response:

Ensure that UPDATE authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the **Tools Customizer Settings** panel.

CCQS024E **An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.**

Explanation:

To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action:

None.

User response:

Ensure that ALTER authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the **Tools Customizer Settings** panel.

CCQS025I **The display options were saved.**

System action:

None.

User response:

No action is required.

CCQS026E **The customization library data set *data_set_name* could not be opened because the requester does not have UPDATE authority on this data set.**

Explanation:

Users must have UPDATE authority to open the customization library data set. Users must have UPDATE authority to open the customization library data set.

System action:

None.

User response:

Ensure that UPDATE authority for the specified customization library data set is granted or specify a different high-level qualifier for the customization library data set on the **Tools Customizer Settings** panel.

CCQS027E **The customization library data set *data_set_name* could not be created because the requester does not have ALTER authority on this data set.**

Explanation:

To create the customization library data set, ALTER authority on the data set must be granted.

System action:

Processing stops.

User response:

Ensure that ALTER authority for the specific customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the **Tools Customizer Settings** panel.

CCQS029E **The customization library data set is not valid. Enter a valid data set name or use the Tools Customizer default: *data_set_name*.**

Explanation:

The specified data set is invalid.

System action:

Processing stops.

User response:

Specify a valid data set name.

CCQS030E **The following command is not a valid CREATE statement: *command_statement*.**

Explanation:

The specified CREATE command statement is invalid because it contains blanks or alphabetic characters.

System action:

Processing stops.

User response:

Specify a valid CREATE command statement. The correct syntax is CREATE *nn*, where *nn* is 1 - 99.

CCQS031E **The following command is not a valid CREATE statement: *command_statement*. The number that can be specified with the CREATE command is 1 - 99.**

Explanation:

The specified CREATE command statement is invalid because it contains either 0 or a number greater than 99.

System action:

Processing stops.

User response:

Specify a valid CREATE command statement. The correct syntax is CREATE *nn*, where *nn* is 1 - 99.

CCQS033E **A user profile cannot be copied into the same user profile**

Explanation:

The specified data set cannot be copied into user's own user profile.

System action:

Processing stops.

User response:

Enter a different data set name.

CCQS034E **The shared user profile data set *data_set_name* could not be created because the requester does not have UPDATE authority on this data set or because the data set already exists in another volume serial.**

Explanation:

To create a shared user profile data set, the requester must have update authority on the data set, and the specified data set name must be unique.

System action:

Processing stops.

User response:

Ensure that the requester has UPDATE authority on the data set and ensure that the data set name is unique.

CCQS035E **The specified data set already has a user profile. Specify a different data set, or press Enter again to replace the existing user profile.**

Explanation:

Pressing Enter overwrites the previous user profile for the specified data set with user's own user profile.

System action:

Processing stops.

User response:

Specify a different data set name.

CCQS036E **The customization library *data_set_name* already exists in volume and cannot be created in a different volume. Enter a different customization library name.**

Explanation:

The same data set name cannot exist in a different volume.

System action:

Processing stops.

User response:

Specify a different data set name.

CCQS037E **The data set name was either not specified or invalid.**

Explanation:

The data set name specified does not follow the IBM data set name convention.

System action:

Processing stops.

User response:

Specify a valid data set name.

CCQS038E **The specified data set cannot be used.**

Explanation:

The specified data sets contain information that supports Tools Customizer, but this data set cannot be used.

System action:

Processing stops.

User response:

Specify a different data set.

CCQS039E **The specified data set has an invalid record format.**

Explanation:

The specified data set should be saved as a different record format. For example, the record format should be FB (Formatted Block) but it is set to VB (Variable Block).

System action:
Processing stops.

User response:
Specify a valid record format.

CCQT000I **The product configuration ID copied_configuration_ID was successfully copied from configuration_ID.**

Explanation:
The specified configuration ID was copied.

System action:
None.

User response:
No action is required.

CCQT001E **The command_name line command was specified more than once, which is not allowed.**

Explanation:
The specified line command cannot be specified more than one time.

System action:
Processing stops.

User response:
Specify the line command only once.

CCQT002E **The configuration_ID configuration ID already exists. Specify a different configuration ID.**

Explanation:
The specified configuration ID exists.

System action:
Processing stops.

User response:
Ensure that the specified configuration ID is unique.

CCQT003I **The product configuration ID configuration_ID was created.**

Explanation:
The specified configuration ID was created.

System action:
None.

User response:
No action is required.

CCQT004I **The product configuration ID configuration_ID was removed.**

Explanation:

The specified configuration ID was removed.

System action:
None.

User response:
No action is required.

CCQT005E **The product configuration ID configuration_ID is not valid. The product configuration ID cannot contain a colon (:).**

Explanation:
The specified configuration ID contains a colon (:), but a colon is not valid.

System action:
Processing stops.

User response:
Specify a configuration ID that does not contain a colon.

CCQT006E **The configuration_ID configuration ID exists. Specify a different configuration ID.**

Explanation:
The specified configuration ID exists.

System action:
Processing stops.

User response:
Specify another configuration ID.

CCQT007E **The configuration_ID configuration ID exists but was removed from the list of configurations. To use this configuration ID, you must restore it.**

Explanation:
The specified configuration ID exists but was removed from the list of available configuration.

System action:
Processing stops.

User response:
Specify another configuration ID. To restore the specified configuration ID, issue the CREATE command, and specify the same configuration ID again.

CCQT008E **The configuration_ID configuration ID exceeds maximum_number characters.**

Explanation:
The specified configuration ID contains too many characters.

System action:
Processing stops.

User response:

Specify another configuration ID that does not exceed the maximum number of characters that was set by Db2 Cloning Tool.

CCQT010I **Create request for *configuration_ID* configuration was cancelled by user.**

Explanation:

The request to create the specified configuration was canceled.

System action:

Processing stops.

User response:

No action is required.

CCQT011I **The *configuration_ID* configuration was not copied.**

Explanation:

The specified configuration was not copied.

System action:

Processing stops.

User response:

No action is required.

CCQT012I **The *configuration_ID* configuration was not removed.**

Explanation:

The specified configuration was not removed.

System action:

Processing stops.

User response:

No action is required.

CCQT013I **None of the configurations were copied or removed. All of the previously selected configurations are deselected.**

Explanation:

The selected configurations were not copied or removed, and they are deselected.

System action:

Processing stops.

User response:

No action is required.

CCQT014E **Specify Y or N and press Enter to continue, or press End to cancel.**

Explanation:

A function requires input.

System action:

Processing stops.

User response:

To continue, specify Y or N and press Enter. Otherwise, press End to cancel.

CCQT015E **The *command_name* command is not allowed during the process of "Select" configuration line command.**

Explanation:

The specified command is not allowed while the line command for selecting configurations is processing.

System action:

Processing stops.

User response:

Remove the specified line command.

CCQT016I **The *configuration_ID* configuration was not created**

Explanation:

The specified configuration was not created.

System action:

Processing stops.

User response:

No action is required.

CCQT017I **The *configuration_ID* configuration was not copied.**

Explanation:

The specified configuration was not copied.

System action:

Processing stops.

User response:

No action is required.

CCQT018E **Specify Y or N, and press Enter.**

Explanation:

A function requires input.

System action:

Processing stops.

User response:

To continue, specify Y or N, and press Enter.

CCQT019I **The select *configuration_ID* configuration process ended.**

Explanation:

The select process for the specified configuration is finished.

System action:

Processing stops.

User response:

No action is required.

CCQT020E **The *configuration_ID* configuration was not created because the data store was not accessible.**

Explanation:

The specified configuration was not created because the data store could not be accessed.

System action:

Processing stops.

User response:

Ensure that the data store is accessible and create the configuration again.

CCQT021E **The *configuration_ID* configuration was not copied because the data store was not accessible.**

Explanation:

The specified configuration was not copied because the data store could not be accessed.

System action:

Processing stops.

User response:

Ensure that the data store is accessible and copy the configuration again.

CCQT025I **The *configuration_ID* configuration was not updated.**

Explanation:

The specified configuration was not updated because the edit process was canceled.

System action:

Processing stops.

User response:

No action is required.

CCQT027I **The product configuration ID has been updated from *edit_from_id* to *edit_to_id*.**

System action:

Processing continues.

User response:

No action is required.

CCQT028I **The product configuration ID has been updated from *edit_from_id* to *edit_to_id*, and the description has been updated from *edit_from_des* to *edit_to_des*.**

System action:

Processing continues.

User response:

No action is required.

CCQT029I **The product configuration description has been updated from *edit_from_des* to *edit_to_des*.**

System action:

Processing continues.

User response:

No action is required.

CCQX001S ***Product_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this product.**

Explanation:

The specified product was customized by using values from the specified data store data set.

System action:

Processing stops.

User response:

Use the specified data store data set to continue customizing the product.

CCQX002S ***component_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this component.**

Explanation:

The specified component was customized by using values from the specified data store data set.

System action:

Processing stops.

User response:

Use the specified data store data set to continue customizing the component.

CCQX011I ***Product_name* was not found.**

Explanation:

The specified product was not found.

System action:

Processing stops.

User response:

Specify another product.

Db2 Cloning Tool messages

Use the information in these messages to help you diagnose and solve Db2 Cloning Tool problems.

CKZ002E **Insufficient region size. Available region size of at least 50000 is required**

Explanation:

The available region size is insufficient to run the Db2 Cloning Tool ISPF interface.

User response:

Contact your systems administrator to increase the region size.

CKZ003E **Must be numeric**

Explanation:

An invalid value was entered.

User response:

Enter a valid numeric value in the field.

CKZ004E **Invalid Value - You must enter a "Y" or "N"**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ005E **Invalid Value - You must enter a "/", "Y", "N" or blank**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ006E **Invalid Value - You must enter a "YES", "NO" or blank**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ007E **Multi selection is not available**

Explanation:

More than one element was selected from the list.

User response:

Choose only one element from the list.

CKZ008E **Invalid Value - You must enter a numeric value from *from value* to *to value***

Explanation:

An invalid value was entered.

User response:

Enter a value that is between the two values listed provided in the message text.

CKZ009E **Invalid Value - You must enter a *valid value***

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ010I **No objects match the filter**

Explanation:

The list of objects is empty. There are no objects that meet the filter criteria.

User response:

Change the filter values or add new objects to the list.

CKZ011I **Operation completed successfully**

Explanation:

The operation successfully completed.

User response:

No action is required.

CKZ012E **The profile with given name already exists**

Explanation:

The name you provided for the profile already exists.

User response:

Enter a different unique profile name.

CKZ013E **The source member list does not match the target member list**

Explanation:

The number of selected sources does not equal the number of selected target subsystems.

User response:

Provide a valid target member for each source member.

CKZ014E **You do not have access rights to this profile**

Explanation:

An attempt was made to edit a profile that was created with no access rights by another user ID. Access to the profile is denied.

User response:

Select a different profile to edit.

CKZ015E **You only have rights to view this profile**

Explanation:

An attempt was made to edit a profile that was created with view only rights by another user ID. Editing the profile is not allowed.

User response:

Use the V line command to view the profile.

CKZ016E **Job Template Variable has invalid name**

Explanation:

An invalid Job Template Variable name was entered. The variable name entered duplicates a reserved variable name.

User response:

Enter a unique Job Template Variable name.

CKZ017E You cannot specify the PARTLEVEL keyword with the RI keyword

Explanation:

A value was entered in the Partlevel field when the RI field was set to YES. This combination is not allowed.

User response:

Either change RI to NO or remove the value from the Partlevel field.

CKZ018E You must specify a valid DD for XML object definition on the DD specification panel

Explanation:

An invalid or an empty value for the XML object definition DD name was entered.

User response:

Enter a valid value for the XML object definition DD name on the DB2 tablespace clone DD Specification panel.

CKZ019E Error during data set creation for XML object definition

Explanation:

An error occurred when creating the data set for XML object definition.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ020E The data set specified for XML object definition already exists, but is not a PDS

Explanation:

The data set specified in the table space cloning DD definition panel already exists, but it is not a PDS.

User response:

Return to the DB2 tablespace clone DD Specification and specify a PDS for the XML DD.

CKZ021E You should fill in a DD for all set DD Names

Explanation:

Not all required DD names have been specified.

User response:

Enter a DD for each required DD name.

CKZ022E Invalid Value - You must enter "SOURCE", "YES", or "NO"

Explanation:

An invalid value was entered in the PRIMARY field.

User response:

Enter SOURCE to specify that the primary BSDS will remain as the primary, as it is on the source subsystem. Enter YES to specify that this member will be the primary member. Enter NO to specify that this member will not be the primary member.

CKZ023E Invalid Value - You must enter "SOURCE" or "NO"

Explanation:

An invalid value was entered in the PRIMARY field.

User response:

Enter SOURCE to specify that the primary BSDS will remain as the primary, as it is on the source subsystem. Enter NO to specify that this member will not be the primary member.

CKZ024E No more than one entry can have PRIMARY set to "YES"

Explanation:

YES was entered in the PRIMARY field for more than one subsystem. Only one subsystem can be specified as the primary member.

User response:

Remove the duplicate value from one of the subsystems.

CKZ025E All entries cannot have PRIMARY set to "NO"

Explanation:

One of the subsystems must be specified as the primary member.

User response:

Enter YES or SOURCE next to one of the subsystems.

CKZ026E Please select surviving target member(s)

Explanation:

A surviving target member must be selected.

User response:

Select at least one surviving target member.

CKZ027E Invalid Value - Please enter a DB2 subsystem ID

Explanation:

An invalid DB2 subsystem ID was entered.

User response:

Specify a valid DB2 subsystem ID.

CKZ028E **Invalid Value - Please enter a data set name**

Explanation:

An invalid data set name was entered.

User response:

Enter a valid data set name in the field.

CKZ029E **Invalid Value - Please enter a member name**

Explanation:

An invalid value was entered.

User response:

Enter a valid member name in the field.

CKZ030E **The DD Name *dd name* already exists**

Explanation:

The name you provided for the DD name already exists.

User response:

Enter a different unique DD name.

CKZ031E **Invalid Value - You must enter "IN" or "OUT"**

Explanation:

An invalid value was entered in the DIR field.

User response:

Enter IN to specify that this DD is provided to ADDRSSU as input; enter OUT to specify that the DD is provided as output.

CKZ032E **Invalid Value - You cannot use a reserved DD name**

Explanation:

A reserved DD name was entered.

User response:

Enter a unique DD name that is not reserved.

CKZ033E **Invalid Value - Please enter a valid LISTDEF name**

Explanation:

An invalid LISTDEF name was entered.

User response:

Enter a valid LISTDEF name in the field.

CKZ034I **Job generation was canceled**

Explanation:

The job generation was canceled by the user.

User response:

No action is required.

CKZ035E **Invalid Value - Please enter a valid control HLQ**

Explanation:

An invalid high level qualifier was entered.

User response:

Enter a valid high level qualifier in the field.

CKZ036E **Invalid Value - Please enter "CR", "DB", "TS", "IX", "IS", "TB", or "VC"**

Explanation:

An invalid OBJECT-TRANSLATE object type was entered

User response:

Enter a valid OBJECT-TRANSLATE type in the field, as described in the message text.

CKZ037E **Invalid Value - Please enter a Source Name**

Explanation:

A source name was not entered.

User response:

Enter a source name.

CKZ038E **Invalid Value - Please enter a Target Name**

Explanation:

A target name was not entered.

User response:

Enter a target name.

CKZ039E **Invalid Value - Please enter a System VCAT**

Explanation:

The system VCAT was not specified for subsystem, which is used as target only.

User response:

Either enter a system VCAT for the subsystem, or change the subsystem so it is not a target only subsystem.

CKZ040E **Invalid Value - You must enter "DB", "TS", "TP", "IX", "IP", "SG", or blank Apply To Type for "STOGROUP" Attribute Name**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ041E **Invalid Value - You must enter "DB", "TS", "TB", "IX", or blank Apply To Type for "CREATEDBY" Attribute Name**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ042E Invalid Value - You must enter "TS", "TP", "IX", "IP", or blank Apply To Type for "GBPCACHE" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ043E Invalid Value - Please enter a valid Source Value for "GBPCACHE" Attribute Name and "TP" Apply To Type: "CHANGED", "ALL", "SYSTEM", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ044E Invalid Value - Please enter a valid Target Value for "GBPCACHE" Attribute Name and "TP" Apply To Type: "CHANGED", "ALL", "SYSTEM", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ045E Invalid Value - Please enter a valid Source Value for "GBPCACHE" Attribute Name and "IP" Apply To Type: "CHANGED", "ALL", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ046E Invalid Value - Please enter a valid Target Value for "GBPCACHE" Attribute Name and "IP" Apply To Type: "CHANGED", "ALL", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ047E Invalid Value - You must enter "TS" or blank Apply To Type for "LOG" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ048E Invalid Value - Please enter a valid Source Value for "LOG" Attribute Name: "YES", "NO", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ049E Invalid Value - Please enter a valid Target Value for "LOG" Attribute Name: "YES", "NO", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ050E Invalid Value - You must enter "TS", "TP", "IX", "IP", or blank Apply To Type for "PRIQTY" and "SECQTY" Attribute Names

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ051E Invalid Value - Please enter a valid Source Value for "PRIQTY" Attribute Name: 1-7 decimal digits, -1, or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ052E Invalid Value - Please enter a valid Target Value for "PRIQTY" Attribute Name: 1-7 decimal digits, -1, or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ053E Invalid Value - Please enter a valid Source Value for "SECQTY" Attribute Name: 1-7 decimal digits, -1, or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ054E Invalid Value - Please enter a valid Target Value for "SECQTY" Attribute Name: 1-7 decimal digits, -1, or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ055E Invalid Value - You must enter "TS", "TP", or blank Apply To Type for "TRACKMOD" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ056E Invalid Value - Please enter a valid Source Value for "TRACKMOD" Attribute Name: "YES", "NO", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ057E Invalid Value - Please enter a valid Target Value for "TRACKMOD" Attribute Name: "YES", "NO", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ058E Invalid Value - You must enter "TS", "IX", or blank Apply To Type for "CLOSE" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ059E Cannot open *profile_version* version profile. The profile was saved with the latest version of the product. Please install all product updates or create a new profile.

Explanation:

You cannot open the profile saved with the latest version of the product.

User response:

Please install all product updates or create a new profile.

CKZ060E Invalid Value - Please enter a valid Source Value for "CLOSE" Attribute Name: "YES", "NO", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ061E Invalid Value - Please enter a valid Target Value for "CLOSE" Attribute Name: "YES", "NO", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ062E Invalid Value - You must enter "TB" or blank Apply To Type for "DATACAPTURE" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ063E Invalid Value - Please enter a valid Source Value for "DATACAPTURE" Attribute Name: "CHANGES", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ064E Invalid Value - Please enter a valid Target Value for "DATACAPTURE" Attribute Name: "CHANGES", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ065E Invalid Value - You must enter "DB", "TS", "IX", or blank Apply To Type for "BUFFERPOOL" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ066E **Invalid Value - Please enter a valid pool name in Source Value for "BUFFERPOOL" Attribute Name or blank**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ067E **Invalid Value - Please enter a valid pool name in Target Value for "BUFFERPOOL" Attribute Name**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ068E **Invalid Value - Please enter a valid Target Value for "CREATEDBY" Attribute Name or blank**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ069E **The Db2 subsystem with given name already exists**

Explanation:

The Db2 subsystem with the name that was provided already exists.

User response:

Enter another Db2 subsystem name.

CKZ070I **The *product_header* is not available**

Explanation:

This message is informational.

User response:

No action is required.

CKZ071I **The *product_header* is not available**

Explanation:

This message is informational.

User response:

No action is required.

CKZ072I **The *product_header* is not available**

Explanation:

This message is informational.

User response:

No action is required.

CKZ073I **The *product_header* is not available**

Explanation:

This message is informational.

User response:

No action is required.

CKZ074I **The *product_header* is not available**

Explanation:

This message is informational.

User response:

No action is required.

CKZ075I **The *product_header* is not available**

Explanation:

This message is informational.

User response:

No action is required.

CKZ076I **The specified SSID is not defined. Please provide the information for it.**

Explanation:

This message is informational.

User response:

No action is required.

CKZ077I **Source Db2 SSID was changed from *subsystem_id* to *subsystem_id***

Explanation:

This message is informational.

User response:

No action is required.

CKZ090E **Invalid Value - Enter a "YES" or "NO"**

Explanation:

An invalid value was entered.

User response:

Enter a valid value. Valid values are "YES" or "NO."

CKZ091E **The INCREMENTAL keyword cannot be used with FCNOCOPY keyword**

Explanation:

Two mutually exclusive keywords were specified.

User response:

Change the value of either the INCREMENTAL field or the FCNOCOPY field.

CKZ092E **If FCSETGTOK is "YES", then FCNOCOPY must also be specified**

Explanation:

A required keyword was not specified.

User response:

Change the value of either the FCNOCOPY field or the FCSETGTOK field.

CKZ093E **Invalid Value - Enter one of the following valid values: "NO", "YES", "PRESMIRREQ", "PRESMIRPREF" or "PRESMIRNONE"**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ094E **The Db2 subsystem with specified SSID is not defined in the Db2 control file. Use the *command* command to create the Db2 subsystem profile**

Explanation:

The SSID does not match any existing Db2 subsystems in the Db2 control file.

User response:

Either create a Db2 subsystem with the specified SSID or specify the name of an existing Db2 subsystem.

CKZ095E **The TARGET-JOB-REPAIR-DDN and TARGET-JOB-INDEX-REBUILD-DDN values cannot be the same**

Explanation:

The TARGET-JOB-REPAIR-DDN and TARGET-JOB-INDEX-REBUILD-DDN values are the same, but they must be different.

User response:

Enter a different TARGET-JOB-INDEX-REBUILD-DDN or TARGET-JOB-REPAIR-DDN value.

CKZ096I **The type of cloning was changed to ONLINE because System Level Backup Type was set to DB2SLB**

Explanation:

Cloning from an SLB does not support offline cloning. The type of cloning was changed to ONLINE for all pairs of Db2 subsystems.

User response:

No action is required.

CKZ101E **ISPF error : *error text***

Explanation:

An ISPF error occurred and caused the displayed message to be generated.

User response:

Correct the error and retry. If necessary, review the ISPF documentation to determine the cause of the error.

CKZ102E **Invalid Command - Please enter a valid command**

Explanation:

An invalid command was entered in the Command or Option line. Valid commands are listed on the panel.

User response:

Enter a valid command.

CKZ103E **Invalid Line Command - Please enter a valid line command**

Explanation:

An invalid line command was entered. Valid line commands are listed at the top of the panel.

User response:

Enter a valid line command.

CKZ104E **Invalid Selection - Please enter a valid option**

Explanation:

An invalid option was entered.

User response:

Enter a valid option.

CKZ105E **Invalid Value - Enter a valid value**

Explanation:

An invalid value was entered.

User response:

Enter a valid value in the field.

CKZ106I **Move is pending. Enter A(After) or B(Before)**

Explanation:

The M(Move) line command was entered but an A(After) or B(Before) command was not specified.

User response:

Enter the A(After) or B(Before) line command, depending on the desired resulting order.

CKZ107E **Result not found**

Explanation:

The desired element was not found.

User response:

Verify that you correctly typed the desired element.

CKZ108I **Select an element**

Explanation:

No element was selected from the list.

User response:

Select at least one element from the list.

CKZ200E *message text***Explanation:**

An internal error of DB2 control file routine or VSAM data repository routine occurred.

User response:

Consult the user guide for the explanation of error codes from those routines. If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support.

CKZ201E **The Db2 Cloning Tool repository does not exist****Explanation:**

The base PDS is not a repository HLQ.

User response:

Ensure the HLQ variable specified for the VSAM data repository in the CKZCLIST is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ202E **The Db2 Cloning Tool DB2 control file does not exist****Explanation:**

The base PDS is not a DB2 control file HLQ.

User response:

Ensure the HLQ variable specified for the DB2 control file in the CKZCLIST is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ203E **The data sets and members that are specified in the CKZDDL DD for the target TCP/IP server job and in the CKZDDL DD for the source job cannot be the same.****Explanation:**

The data set and member name that are specified in the CKZDDL DD for the source job are the same as the data set and member name specified in the CKZDDL DD for the target TCP/IP server job.

User response:

Specify a different data set and/or member name in the CKZDDL DD for the target TCP/IP server job or in the CKZDDL DD for the source job.

CKZ203W **Masking validation routine is not found****Explanation:**

The module CKZ00991 is not found.

User response:

Check that the module CKZ00991 is available in the ISPLLIB. Contact your system administrator to correct the error.

CKZ204E **Unknown error in masking validation routine****Explanation:**

An unknown error occurred in the masking validation routine.

User response:

Contact IBM Software Support.

CKZ301E **Invalid data set name****Explanation:**

An invalid data set name was entered

User response:

Enter a valid data set name in the field.

CKZ302E **Data set is not partitioned****Explanation:**

The specified data set already exists, but it is not partitioned.

User response:

Enter the name of either a partitioned data set or a non-existent data set.

CKZ303E **Cannot open file *filename*****Explanation:**

The file name listed in the message cannot be opened.

User response:

If unable to determine the reason the file open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ304E **Cannot get text from skeleton****Explanation:**

Text could not be read from a skeleton.

User response:

If unable to determine the reason that text cannot be retrieved from skeleton, contact IBM Software Support. Have available the listing that contains this message.

CKZ305E **Error during generation of COPY JCL****Explanation:**

An error occurred when generating the COPY JCL.

User response:

If unable to determine the cause of the failure to generate COPY JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ306E **Error during generation of RENAME JCL****Explanation:**

An error occurred when generating the RENAME JCL.

User response:

If unable to determine the cause of the failure to generate RENAME JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ307E Error during generation of UPDATE JCL

Explanation:

An error occurred when generating the UPDATE JCL.

User response:

If unable to determine the cause of the failure to generate UPDATE JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ308E Error during generation of START JCL

Explanation:

An error occurred when generating the START JCL.

User response:

If unable to determine the cause of the failure to generate START JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ309E Error during generation of STOP JCL

Explanation:

An error occurred when generating the STOP JCL.

User response:

If unable to determine the cause of the failure to generate STOP JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ310E Error during generation of COPYCHECK JCL

Explanation:

An error occurred when generating the COPYCHECK JCL.

User response:

If unable to determine the cause of the failure to generate COPYCHECK JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ311E Error during generation of SQL JCL

Explanation:

An error occurred when generating the SQL JCL.

User response:

If unable to determine the cause of the failure to generate SQL JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ312E Error during generation of FIX JCL

Explanation:

An error occurred when generating the FIX JCL.

User response:

If unable to determine the cause of the failure to generate FIX JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ313E Error during generation of BCSCLEAN JCL

Explanation:

An error occurred when generating the BCSCLEAN JCL.

User response:

If unable to determine the cause of the failure to generate BCSCLEAN JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ314E Error during generation of SETLOG SUSPEND

Explanation:

An error occurred when generating the SETLOG SUSPEND JCL.

User response:

If unable to determine the cause of the failure to generate SETLOG SUSPEND JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ315E Error during generation of SETLOG RESUME JCL

Explanation:

An error occurred when generating the SETLOG RESUME JCL.

User response:

If unable to determine the cause of the failure to generate SETLOG RESUME JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ316E Error during generation of DB2RBLDBSDS JCL

Explanation:

An error occurred when generating the DB2RBLDBSDS JCL.

User response:

If unable to determine the cause of the failure to generate DB2RBLDBSDS JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ317E Error during generation of DB2LGRNXCLEAN JCL

Explanation:

An error occurred when generating the DB2LGRNXCLEAN JCL.

User response:

If unable to determine the cause of the failure to generate DB2LGRNXCLEAN JCL, contact IBM Software

Support. Have available the listing that contains this message.

CKZ318E Error during generation of DB2XCFCLEAN JCL

Explanation:

An error occurred when generating the DB2XCFCLEAN JCL.

User response:

If unable to determine the cause of the failure to generate DB2XCFCLEAN JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ319I JCL was successfully generated

Explanation:

The JCL for the profile was successfully generated.

User response:

No action is required.

CKZ320E Error during data set creation for JCL

Explanation:

An error occurred when creating the data set for the JCL.

User response:

If unable to determine the reason the creating failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ321E Error generating source JCL

Explanation:

An error occurred when generating the source JCL.

User response:

If unable to determine the cause of the failure to generate Source JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ322E Error generating target JCL

Explanation:

An error occurred when generating the target JCL.

User response:

If unable to determine the cause of the failure to generate target JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ323E Error generating TCP/IP server JCL

Explanation:

An error occurred when generating the TCP/IP server JCL.

User response:

If unable to determine the cause of the failure to generate TCP/IP server JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ324E Error during generation of MASKDEF

Explanation:

An error occurred when generating the MASKDEF JCL.

User response:

If unable to determine the cause of the failure to generate MASKDEF JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ325E Error during generation of LISTDEF

Explanation:

An error occurred when generating the LISTDEF JCL.

User response:

If unable to determine the cause of the failure to generate LISTDEF JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ326E Invalid Value - You must enter "YES" or "NO"

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ327E You must specify special ZPARMS for DB2 subsystem *subsystem ID* before generating jobs

Explanation:

The ZPARMS field was not specified for this DB2 subsystem before job generation.

User response:

Edit the specified DB2 subsystem under Administrator Functions on the main menu. Provide the ZPARMS member name for the subsystem.

CKZ328E DB2 subsystem *subsystem ID* not found

Explanation:

The subsystem with the listed SSID is not found in the list of subsystems.

User response:

Verify that you correctly entered the SSID, or select another DB2 subsystem.

CKZ329E You must specify load library data sets for DB2 subsystem *subsystem ID* before generating jobs

Explanation:

The load library was not defined for the subsystem specified in the message.

User response:

Edit the specified DB2 subsystem under Administrator Functions on the main menu. Provide at least one load library for the subsystem.

CKZ330E File tailoring open returned a file tailoring already in progress condition

Explanation:

An attempt to perform file tailoring for utility customization failed. There was a file tailoring session already in progress. File tailoring sessions cannot be performed concurrently.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ331E File tailoring open returned the output file already in use condition -- ENQ failed

Explanation:

An attempt to open the DB2 control data set failed with an ENQ error. The data set is already open for output.

User response:

Verify that you are the only user attempting to access this file.

CKZ332E File tailoring open returned the skeletal file or output file not allocated condition

Explanation:

An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response:

An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

CKZ333E File tailoring open returned a severe error condition

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on open.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ334E File tailoring open returned an unknown code -- severe error

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on open.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ335E File tailoring close returned a file not open condition -- severe error

Explanation:

An attempt to perform file tailoring failed because a File-Not-Open condition was encountered on close.

User response:

Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.

CKZ336E File tailoring close returned an output file in use condition

Explanation:

An attempt to perform file tailoring failed because an Output-File-In-Use condition was encountered on close.

User response:

Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.

CKZ337E File tailoring close returned a skeletal file or output file not allocated condition

Explanation:

An attempt to close file tailoring failed because either a tailoring skeleton file or output file was not allocated.

User response:

Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.

CKZ338E File tailoring close returned a severe error

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ339E File tailoring close returned an unknown code -- severe error

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ340E File tailoring close returned an output member exists in the

output library and NOREPL was specified

Explanation:

An attempt to perform file tailoring failed because the close process could not replace the pre-existing tailored member in the output file.

User response:

Change the output member name to a new name or ensure that the output library allows for member replacement.

CKZ341E File tailoring include returned a skeleton does not exist condition

Explanation:

An attempt to perform file tailoring failed because the tailoring process could not locate a required tailoring skeleton.

User response:

Ensure that all required files are allocated to perform file tailoring.

CKZ342E File tailoring include returned a skeleton in use -- ENQ failed condition

Explanation:

An attempt to access a tailoring skeleton failed with an ENQ error (member in use).

User response:

Verify that all required tailoring files are allocated and that there are no other tailoring sessions running concurrently.

CKZ343E File tailoring include returned a data truncation or skeleton library or output file not allocated condition

Explanation:

An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response:

Verify that all required files are allocated prior to performing file tailoring.

CKZ344E File tailoring include returned a severe error condition

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ345E File tailoring include returned an unknown condition -- severe error

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response:

Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ346E Allocation Error - An error was encountered allocating the ISPF DD - Process not completed

Explanation:

An error was encountered allocating the ISPF DD.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ347E Allocation Error - An error was encountered reading the ISPF DD - Process not completed

Explanation:

An error was encountered reading the ISPF DD.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ351W Masking validation routine is not found

Explanation:

The module CKZ00991 is not found.

User response:

Check that the module CKZ00991 is available in ISPLLIB. Contact your system administrator to correct this error.

CKZ352E *message text*

Explanation:

An internal error of masking routine occurred.

User response:

Consult the user guide for the explanation of error codes from the masking routine. If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support.

CKZ353E Unknown error of masking validation routine

Explanation:

An unknown error was encountered in the masking validation routine.

User response:

Contact IBM Software Support.

CKZ354E DDNAMEs *ddname1* and *ddname2* have the same data set name in Source DD Specification

Explanation:

The same data set name has been specified for different DDs.

User response:

Change one of the data set names.

CKZ355E DDNAMEs *ddname1* and *ddname2* have the same data set name in Target DD Specification

Explanation:

The same data set name has been specified for different DDs.

User response:

Change one of the data set names.

CKZ356E DDNAMEs *ddname1* and *ddname2* have the same data set name in TCPIP Server DD Specification

Explanation:

The same data set name has been specified for different DDs.

User response:

Change one of the data set names.

CKZ357E The source and target DD specifications are not synchronized. Please go to source job DD specification and synchronize it with target job DD specification

Explanation:

The source and target DD specifications are not synchronized.

User response:

Please go to source DD specification and synchronize it with target DD specification.

CKZ358E This attribute is duplicated

Explanation:

More than one attribute with the same value was specified. The attribute value must be unique.

User response:

Change one of the attribute values.

CKZ360E You must specify LOCAL-SSID for the SET command before generating jobs

Explanation:

A LOCAL-SSID was not specified for the SET command.

User response:

Edit the profile and enter a LOCAL-SSID in the field for the SET command before generating jobs.

CKZ361E You must specify TARGET-DB2 SSID for the COPY command before generating jobs

Explanation:

A TARGET-DB2 SSID was not specified for the COPY command.

User response:

Edit the profile and enter a TARGET-DB2 in the field for the COPY command before generating jobs.

CKZ362E You must specify LOCAL-SSID for the TCPIP Server SET command before generating jobs

Explanation:

A LOCAL-SSID was not specified for the TCPIP Server SET command.

User response:

Edit the profile and enter a LOCAL-SSID in the field for the TCPIP Server SET command before generating jobs.

CKZ363E There is no data sharing group to satisfy the number of members for targets

Explanation:

There are no data sharing groups that can be used as targets, because none of the potential target data sharing groups have enough members.

User response:

Add the required number of members into an existing target data sharing group or create the new target data sharing group with enough members to satisfy the selected source.

CKZ364E There are not enough available members for targets in the target data sharing group

Explanation:

There are no data sharing groups that can be used as targets, because none of the potential target data sharing groups have enough members.

User response:

Add more target member in the selected target data sharing group or select source members for cloning from another data sharing group

CKZ365E **You must specify System VCAT for DB2 subsystem *subsystem ID* before generating jobs**

Explanation:

The system VCAT was not specified for the DB2 subsystem described in the message text.

User response:

Enter a system VCAT for the DB2 subsystem described in the message text before generating jobs.

CKZ366E **You must specify all table creators for the MASKDEF Command or the DEFAULT-SQLID for SET Command before generating jobs**

Explanation:

Either the MASKDEF command did not specify all table creators, or the DEFAULT-SQLID field for the SET command was not filled in.

User response:

Either specify all table creators for MASKDEF Command, or specify DEFAULT-SQLID for the SET command.

CKZ367E **Target DD *ddname* has different data set name from source. Please correct the target DD or synchronize Target and Source DD Specifications**

Explanation:

Different DDs were specified for the same DD name on the **Db2 tablespace clone DD Specification** panel and the **Setup Target DD Cards** panel.

User response:

Change the target DD of the DD name that is shown in the message so that both DDs are the same in the **Db2 tablespace clone DD Specification** panel and in the **Setup Target DD Cards** panel.

CKZ368E **You cannot use the same DDNAME *ddname1* in JOB-TEMPLATE and DD Specification**

Explanation:

The same DD names were defined in the JOB-TEMPLATE and the DD Specification.

User response:

Change or remove one of the DD names.

CKZ369E **You cannot use the same data set name in JOB-TEMPLATE and DDNAME *ddname1* of DD Specification**

Explanation:

The same data set names were defined in the JOB-TEMPLATE and one of the DD names in the DD Specification.

User response:

Change one of data set names or remove a DD name from the DD Specification.

CKZ371E **You must specify *dd_name* in Db2 tablespace DD Specification before generating jobs**

Explanation:

The *dd_name* that is listed in the message must be specified in the **Db2 tablespace DD Specification** panel before generating jobs.

User response:

Specify the *dd_name* in the **Db2 tablespace DD Specification** panel.

CKZ372E **You must select *dd_name* in Db2 tablespace DD Specification before generating jobs**

Explanation:

The *dd_name* that is listed in the message must be selected on the **Db2 tablespace DD Specification** panel before generating jobs.

User response:

Select the *dd_name* on the **Db2 tablespace DD Specification** panel.

CKZ373E **You must select *dd_name* in Db2 tablespace clone DD Specification before generating jobs**

Explanation:

The *dd_name* that is listed in the message must be selected on the **Db2 tablespace clone DD Specification** panel before generating jobs.

User response:

Select the *dd_name* on the **Db2 tablespace clone DD Specification** panel.

CKZ374E **Error during generation of DB2GETBACKINFO JCL**

Explanation:

An error occurred when generating DB2GETBACKINFO JCL.

User response:

If you are unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ375E **Error during generation of BACKINFO-REFORMAT JCL**

Explanation:

An error occurred when generating BACKINFO-REFORMAT JCL.

User response:

If you are unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ376E Error during generation of COPY PGM(NONE) JCL

Explanation:

An error occurred when generating COPY PGM(NONE) JCL.

User response:

If you are unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ378E Error during generation of CKZRNTGT EXEC JCL

Explanation:

An error occurred when generating CKZRNTGT EXEC JCL.

User response:

If you are unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ379E Error during generation of VOLOPTIONS NEWTARGETS JCL

Explanation:

An error occurred when generating VOLOPTIONS NEWTARGETS JCL.

User response:

If you are unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ380E *symbol1* is not supported

Explanation:

An invalid value was entered. This combination of field values is not allowed.

User response:

Change the value of at least one field. If the object type is DATABASE or STOGROUP, changed the Object Specification Qualifier 1 field. Otherwise, change the value of either the Object Specification Qualifier 1 field or the Object Specification Qualifier 2 field.

CKZ381E You must enter CATWORK-DSN MASK in Db2 subsystem COPY command because ICF catalogs are specified

Explanation:

CATWORK-DSN MASK is required if ICF catalogs are specified.

User response:

Enter a valid CATWORK-DSN MASK value.

CKZ382E Invalid Value - Please enter a valid CATWORK-DSN MASK value

Explanation:

An invalid CATWORK-DSN MASK value was entered.

User response:

Enter a valid CATWORK-DSN MASK value.

CKZ383E RENAME TYPE must be SAFE if VOLBKUP-DDN parameter is specified

Explanation:

RENAME TYPE must be SAFE because the VOLBKUP-DDN parameter is specified.

User response:

Enter SAFE for the RENAME TYPE.

CKZ384E Error on file *filename* I/O Handler: error message

Explanation:

An error occurred that is documented in the message.

User response:

Correct the error described in the message and retry.

CKZ385E You must specify bootstrap data sets for Db2 subsystem *subsystem_id* before generating job

Explanation:

The bootstrap data sets were not defined for the subsystem that is specified in the message.

User response:

Edit the specified Db2 subsystem under Administrator Functions on the **Primary Option Menu** panel. Provide both bootstrap data sets for the subsystem.

CKZ386E You must specify ZPARMs member for Db2 subsystem *subsystem_id* before generating job

Explanation:

The ZPARM member name was not defined for the subsystem specified in the message.

User response:

Edit the specified Db2 subsystem under Administrator Functions on the **Primary Option Menu** panel. Provide the ZPARM member name for the subsystem.

CKZ387E The DD name *dd_name* already exists in product DD list

Explanation:

The DD with the name that you provided for *dd_name* already exists.

User response:

Enter a unique DD name.

CKZ388E **The DD name *dd_name* already exists in User DD list**

Explanation:

The DD with the name that you provided for *dd_name* already exists.

User response:

Enter a unique DD name.

CKZ389E **You must specify *MINILOG-HLQ* for *LOG-APPLY* command if *LA-ENABLE* is set to "YES"**

Explanation:

The *MINILOG-HLQ* field for the *LOG-APPLY* command is empty and *LA-ENABLE* is set to "YES."

User response:

Enter the *MINILOG-HLQ* for the *LOG-APPLY* command using the **Set *MINILOG* options** panel.

CKZ390E **The *SUBTASK-DATASET-EXTENSIONS* cannot be enabled when *parameter* for the *command* command is set to "YES"**

Explanation:

SUBTASK-DATASET-EXTENSIONS(YES) is not allowed if the *parameter* keyword is set to "YES" for the *command* command.

User response:

Change the value of either the *SUBTASK-DATASET-EXTENSIONS* field or the *parameter* field for the *command* command.

CKZ391E **The *parameter* cannot be enabled when *SUBTASK-DATASET-EXTENSIONS* for the *SET* Command is set to "YES"**

Explanation:

parameter (YES) is not valid if the *SUBTASK-DATASET-EXTENSIONS* keyword for the *SET* command is set to "YES."

User response:

Change the value of either the *SUBTASK-DATASET-EXTENSIONS* field for the *SET* command or the *parameter* field.

CKZ392E **The *SUBTASK-DATASET-EXTENSIONS* cannot be enabled when *USE-RUNTIME-REPOSITORY* is set to "YES"**

Explanation:

SUBTASK-DATASET-EXTENSIONS(YES) is not allowed if the *USE-RUNTIME-REPOSITORY* keyword is set to "YES."

User response:

Change the value of either the *SUBTASK-DATASET-EXTENSIONS* field or the *USE-RUNTIME-REPOSITORY* field.

CKZ393E **Invalid Value - Please enter a valid control HLQ in Source DD Specification**

Explanation:

An invalid high level qualifier was entered.

User response:

Enter a valid high level qualifier in the Source DD Specification.

CKZ394E **The target data set *VCAT* must be provided when *COPY-IJ-TO-NONEXISTENT-TARGET* for the *Copy* Command is set to "YES"**

Explanation:

A target data set *VCAT* was not specified, but it is required when *COPY-IJ-TO-NONEXISTENT-TARGET* for the *COPY* command is "YES."

User response:

Either specify a target data set *VCAT* or change value of the *COPY-IJ-TO-NONEXISTENT-TARGET* field for the *COPY* command. For defining the target data set *VCAT*, either use the *OBJECT-TRANSLATE VCAT* command or the target *DEFVCAT* parameter on the **Source and Target Db2 subsystems** panel.

CKZ395E **Invalid Value - Enter a valid Source Mask**

Explanation:

An invalid Source Mask value was entered.

User response

Enter a valid Source Mask value. Syntax rules for source data set rename masks are as follows:

- A single asterisk (*) represents exactly one DSN qualifier of any value. For example: *, or *. or *
- A double asterisk (**) represents 0 to *nn* DSN qualifiers of any value. Double asterisks (**) cannot appear with any other characters within a qualifier. Three or more adjacent asterisks (***) are not allowed within a qualifier. Double asterisks (**) can be used more than once in a source rename mask.
- A percent sign (%) represents one non-blank character.
- An exclamation point (!) represents one national character: @ # \$.
- A less-than sign (<) represents one non-numeric character, national symbols included.
- A greater-than sign (>) represents one numeric character.

For more information on source masks, refer to the “Filtering pattern masks” on page 26 topic.

CKZ396E Invalid Value - Enter a valid Target Mask

Explanation:

An invalid Target Mask value was entered.

User response

Enter a valid Target Mask value for each source mask. You can enter masks so that the target data set names will be exactly the same as the source data set names. Syntax rules for target data set rename masks are as follows:

- A single asterisk (*) represents exactly one DSN qualifier. * cannot be used for a partial qualifier in a target rename mask. For example: aaa.*bb.** is not be valid.
- A double asterisk (**) represents 0 to *nn* DSN qualifiers. Notes: ** cannot appear with any other characters within a qualifier. Three or more adjacent asterisks are not allowed within a qualifier. ** may only be used once in a target rename mask.
- A percent sign (%) represents one non-blank character.
- An exclamation point (!) represents one national character: @ # \$.
- A plus sign followed by 1 to 8 characters (+cccc) means "insert this new qualifier."
- A minus sign (-) means "remove this qualifier from the target data set name."

For more information on target masks, see the “Filtering pattern masks” on page 26 topic.

CKZ397E If Target *class_type* is defined then Source *class_type* must also be defined

Explanation:

The source *class_type* cannot be empty if the target *class_type* is defined.

User response:

Either enter the source *class_type* or remove the target *class_type*.

CKZ398E If Source *class_type* is defined then Target *class_type* must also be defined

Explanation:

Target *class_type* cannot be empty if source *class_type* is defined.

User response:

Either enter a target *class_type* or remove the source *class_type*.

CKZ399E The "*DEFAULT" value can be used as Source *class_type* only once

Explanation:

The "*DEFAULT" value is only allowed in one pair.

User response:

Retain the "*DEFAULT" value only for one pair and replace this value in other pairs.

CKZ400E You must specify source storage groups or source volumes if DATA-MOVER PGM for the COPY command is equal to *program_name*

Explanation:

Source storage groups or source volumes were not specified, but they are required if DATA-MOVER PGM for the COPY command is set to *program_name*.

User response:

Specify source storage groups or source volumes using the **Source and Target Volume Pairing** panel.

CKZ401E You must specify target storage groups or target volumes if DATA-MOVER PGM for the COPY command is equal to *program_name*

Explanation:

Target storage groups or target volumes were not specified, but they are required if DATA-MOVER PGM for the COPY command is set to *program_name*.

User response:

Specify target storage groups or target volumes using the **Source and Target Volume Pairing** panel.

CKZ402E A different number of source and target *element_type* is specified

Explanation:

An equal number of source and target elements must be specified when **Keep volumes sequence** is enabled.

User response:

Either enter an equal number of source and target elements or disable the **Keep volumes sequence** parameter.

CKZ403E You must specify target volumes if "Keep volumes sequence" field is equal to "YES"

Explanation:

Target volumes were not specified, but they are required.

User response:

Either specify target volumes using the **Source and Target Volume Pairing** panel, or change the value of the **Keep volumes sequence** field to specify target storage groups.

CKZ404E **The cloning scenario is not supported when "Keep volumes sequence" is enabled**

Explanation:

An invalid combination of values was entered. This combination of field values is not allowed.

User response:

Refer to the help panel for the active panel for more information.

CKZ405E **You must specify source storage groups if "Pair using Source Storage Group Names" field is equal to "YES"**

Explanation:

Source storage groups were not specified, but they are required.

User response:

Either specify source storage groups using the **Source and Target Volume Pairing** panel, or change the value of the **Pair using Source Storage Group names** field.

CKZ406I **Inconsistent member ID settings for data sharing group**

Explanation:

A member ID should be specified either for all members of the data sharing group or for none of them.

User response:

Either set or clear the "Member ID" field for all members of the data sharing group.

CKZ407I **Member IDs in data sharing group are not unique**

Explanation:

All members in the data sharing group must have a unique member ID.

User response:

Make sure that each member of the data sharing group has a unique member ID.

CKZ408E **Member ID settings for data sharing group *group_name* are not consistent**

Explanation:

A member ID should be specified either for all members of the data sharing group or not specified at all.

User response:

Edit the settings for each member of the specified data sharing group under Administrator Functions from the main menu. Ensure that the member ID is specified either for all members of the data sharing group or for none of them.

CKZ409E **Member IDs in data sharing group *group_name* are not unique**

Explanation:

All members of the data sharing group must have unique member IDs.

User response:

Edit the settings for members of the specified data sharing group under Administrator Functions from the main menu. Ensure that each member of the data sharing group has a unique member ID.

CKZ410I **DATA-MOVER PGM was changed to "SRCIMCPY". Therefore the following values of keywords were set: LA-ENABLE(Y), FUZZY-COPY(Y), DATA-MASKING(N), SUBTASK-DATASET-EXTENSIONS(N), USE-RUNTIME-REPOSITORY(N)**

Explanation

DATA-MOVER PGM was set to SRCIMCPY. To clone from source image copies, log apply is required; therefore, LA-ENABLE for the LOG-APPLY command was set to YES. In addition, settings that are incompatible with cloning from source image copies were modified as follows:

- FUZZY-COPY for the COPY command was set to YES.
- DATA-MASKING for the COPY command was set to NO.
- SUBTASK-DATASET-EXTENSIONS and USE-RUNTIME-REPOSITORY for the SET command were set to NO.

User response:

No action is required.

CKZ411E **Length of this string must be 12 or 20 hexadecimal characters**

Explanation:

The specified value is not valid.

User response:

Enter a valid value.

CKZ412E **You must specify *required_parameter* if END-POINT TYPE is set to *end_point_type***

Explanation:

A required parameter was not specified. The required parameter is listed in the message.

User response:

Enter the required parameter.

CKZ413E **The *parameter* cannot be enabled when DATA-MOVER PGM for the COPY command is set to *program***

Explanation:

The parameter value that is listed in the message is not valid if DATA-MOVER PGM for the COPY command is set to SRCIMCPY or SRCVSCP.

User response:

Change the value of either the parameter or the DATA-MOVER PGM field for the COPY command.

CKZ414E **The TARGET-JOB-INDEX-REBUILD-DDN value must be provided in the COPY command when REBUILD-INDEXES-EXECUTE is set to "YES" in the SET command**

Explanation:

The TARGET-JOB-INDEX-REBUILD-DDN field for the COPY command is not set. This field is required if REBUILD-INDEXES-EXECUTE is set to YES in the SET command.

User response:

Enter a value in the TARGET-JOB-INDEX-REBUILD-DDN field for the COPY command, or set the value of REBUILD-INDEXES-EXECUTE to NO for the Set command.

CKZ415E **Error during generation of RESTORE-FROM-DUMPTAPES JCL**

Explanation:

An error occurred when generating RESTORE-FROM-DUMPTAPES JCL.

User response:

If you are unable to determine the reason for the error from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ416E **Required parameter is not defined**

Explanation:

A required parameter was not specified.

User response:

Specify the required parameter and retry.

CKZ417E **Invalid Value - Enter either a numeric value from 1 to 16 or asterisk (*)**

Explanation:

An invalid value was entered.

User response:

Enter a valid value in the field.

CKZ418E **You must specify target volumes or target storage groups when using the RESTORE-FROM-DUMPTAPES command**

Explanation:

Neither target volumes nor target storage groups were specified, but at least one of them is required.

User response:

Specify either target volumes or target storage groups for the Source and Target Volume Pairing and retry.

CKZ419E **You must specify target storage groups if "Pair using Source Storage Group Names" field is equal to "YES"**

Explanation:

Target storage groups were not specified, but they are required.

User response:

Either specify target storage groups for the Source and Target Volume Pairing or change the value of "Pair using Source Storage Group Names" field and specify target volumes.

CKZ420I **You have already chosen all available object mismatch types**

Explanation:

All available object mismatch types have already been selected.

User response:

No action is required.

CKZ421E **You must specify SERVER-IP for the Source Db2 subsystem if LOG-APPLY is enabled and REMOTE-CONNECT-TYPE is set to *connection_type* in SET command**

Explanation:

The required parameter was not specified.

User response:

Specify the required parameter and retry.

CKZ422E **You must specify SERVER-IP or SERVER-NAME for the Source Db2 subsystem before generating jobs**

Explanation:

The required parameter was not specified.

User response:

Specify SERVER-IP or SERVER-NAME for the source Db2 subsystem using the **Source and Target Db2 subsystems** panel and retry.

CKZ423E **Please enter a Source Db2 subsystem ID**

Explanation:

The source Db2 subsystem ID was not defined.

User response:

Specify the source Db2 subsystem ID using the **Source and Target Db2 subsystems** panel and retry.

CKZ424E Please enter a Target Db2 subsystem ID

Explanation:

The target Db2 subsystem ID was not defined.

User response:

Specify the target Db2 subsystem ID using the **Source and Target Db2 subsystems** panel and retry.

CKZ425E You must specify Source Db2 subsystem ID before generating job(s)

Explanation:

A source Db2 subsystem ID was not specified.

User response:

Specify the source Db2 subsystem ID using the **Source and Target Db2 subsystems** panel and retry.

CKZ426E You must specify Target Db2 subsystem ID before generating job(s)

Explanation:

A target Db2 subsystem ID was not specified.

User response:

Specify the target Db2 subsystem ID using the **Source and Target Db2 subsystems** panel and retry.

CKZ427I The Source TCPIP-STC-NAME was promoted to Source SET command and Source TCPIP Server SET command

Explanation:

The source TCPIP-STC-NAME from the **Source and Target Db2 subsystems** panel was promoted to the same field in the source SET command panel and the source TCPIP Server SET command panel.

User response:

No action is required.

CKZ428I The Target TCPIP-STC-NAME was promoted to Source SET command and TCPIP Server SET command

Explanation:

The target TCPIP-STC-NAME from the **Source and Target Db2 subsystems** panel was promoted to the same field in the source SET command panel and the TCPIP server SET command panel.

User response:

No action is required.

CKZ429I The Source and Target TCPIP-STC-NAMEs were promoted to matched fields in Source SET command, TCPIP Server SET command and Source TCPIP Server SET command

Explanation:

The source TCPIP-STC-NAME from the **Source and Target Db2 subsystems** panel was promoted to the same field in the source SET command panel and the source TCPIP server SET command panel. The target TCPIP-STC-NAME was promoted to the same field in the source SET command panel and the TCPIP server SET command panel.

User response:

No action is required.

CKZ430E Invalid Value - Enter a valid *class_type* name

Explanation:

An invalid *class_type* was entered. The field in which the invalid name was entered is listed in the message text.

User response:

Enter a valid *class_type*. The entered name must be alphanumeric or national (@, #, \$) characters; the first one can be alphabetic or national.

CKZ431E Invalid Value - You must enter a "ALL", "LOB", "BASE", "XML" or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ432E When DATA-MOVER PGM is *program*, the SIM cannot be "ALLOC"

Explanation:

SIM(ALLOC) is not valid if DATA-MOVER PGM is set to SRCIMCPY or SRCVSCPY.

User response:

Change the value of either the SIM field or the DATA-MOVER PGM field.

CKZ433E The number of members in the data sharing group *data_sharing_group* exceeds the maximum value

Explanation:

The maximum number of members in a data sharing group is 32.

User response:

Use Administrator Functions from the main menu to remove some members from the data sharing group that is described in the message text.

CKZ434I DATA-MOVER PGM was changed to "SRCVSCPY". Therefore the following values of keywords were set: LA-ENABLE(Y), SKIP-LOG-APPLY(Y), END-POINT(TO_CURRENT), DATA-MASKING(N), FUZZY-COPY(Y), SUBTASK-DATASET-EXTENSIONS(N), USE-RUNTIME-REPOSITORY(N)

Explanation

DATA-MOVER PGM was set to SRCVSCPY. LOG-APPLY processing is required for SRCVSCPY; therefore, LA-ENABLE for the LOG-APPLY command was set to YES. In addition, settings that are incompatible with SRCVSCPY were modified as follows:

- SKIP-LOG-APPLY for the LOG-APPLY command was set to YES.
- END-POINT for the LOG-APPLY command was set to TO_CURRENT.
- DATA-MASKING for the COPY command was set to NO.
- FUZZY-COPY for the COPY command was set to NO.
- SUBTASK-DATASET-EXTENSIONS and USE-RUNTIME-REPOSITORY for the SET command were set to NO.

User response:

No action is required.

CKZ435W DSS-COPY-COMMANDS * DSNS-PER-COPY = *decimal_number* - may cause storage and other resource-related errors during *program_name* execution. The recommended maximum for DSS-COPY-COMMANDS * DSNS-PER-COPY is 1400

Explanation:

The supplied values for DSS-COPY-COMMANDS and DSNS-PER-COPY multiplied together give a product of *decimal_number*. This might cause storage and other resource-related errors during execution of the program that is listed in the message. The recommended maximum for DSS-COPY-COMMANDS * DSNS-PER-COPY is 1400.

User response:

Change the value of either the DSS-COPY-COMMANDS field or the DSNS-PER-COPY field on the **Db2 tablespace clone COPY Command** panel.

CKZ438E Enable TARGET-JOB-REPAIR-EXECUTE for the SET command and/or specify a value TARGET-JOB-REPAIR-DDN field for the COPY command.

Explanation:

A required parameter was not specified.

User response:

Correct the error that is described in the message and retry.

CKZ440E The Source and Target Db2 subsystems list is empty

Explanation:

The list of source and target Db2 subsystems is empty.

User response:

Enter at least one pair of source and target Db2 subsystems.

CKZ441E You must specify prefix for work data sets before generating jobs

Explanation:

The prefix for work data sets cannot be empty.

User response:

Specify a prefix for work data sets and retry.

CKZ442E You must specify work data sets unit device before generating jobs

Explanation:

The work data sets unit device cannot be empty.

User response:

Specify the work data sets unit device and retry.

CKZ443E Error during generation of *member_name* | *command_name* JCL

Explanation:

An error occurred when generating the JCL.

User response:

If unable to determine the cause of the failure to generate JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ444E You must select surviving target member(s) before generating jobs

Explanation:

A surviving target member for *data_sharing_attribute* group with *group_name* group name must be selected.

User response:

Select at least one surviving target member for *data_sharing_attribute* group with *group_name* group name.

CKZ445E **You must enter a valid Rename mask for RENAME command before generating jobs**

Explanation:

An invalid rename mask was entered.

User response:

Enter a valid rename mask and retry.

CKZ446E **You must specify Rename masks for RENAME command before generating jobs**

Explanation:

A required parameter was not entered.

User response:

Include the missing parameter and retry.

CKZ447E **You must specify Db2 HLQs for source and target subsystem pair with *subsystem_id* Source Subsystem ID and *subsystem_id* Target Subsystem ID before generating jobs**

Explanation:

The source and target Db2 data set high level qualifiers were not defined.

User response:

Provide Db2 high level qualifiers for the source and target subsystem pair that is described in the message and retry.

CKZ448E **If Target ICF Catalog is defined then Source ICF Catalog must also be defined**

Explanation:

A source ICF catalog cannot be empty if a target ICF catalog is defined.

User response:

Either enter a source ICF catalog or remove the target ICF catalog.

CKZ449E **If Source ICF Catalog is defined then Target ICF Catalog must also be defined**

Explanation:

A target ICF catalog cannot be empty if a source ICF catalog is defined.

User response:

Either enter a target ICF catalog or remove the source ICF catalog.

CKZ450E **You must first specify an INCLUDE clause**

Explanation:

LISTDEF must begin with an INCLUDE clause.

User response:

Specify an INCLUDE clause first. You can use the M (Move) line command for this purpose.

CKZ451E **The SUBTASK-DATASET-EXTENSIONS for the SET command and log apply functionality cannot be used at the same time**

Explanation:

SUBTASK-DATASET-EXTENSIONS(YES) for the SET command is not allowed if the LA-ENABLE keyword is set to "YES" for LOG-APPLY in the COPY command.

User response:

Disable either SUBTASK-DATASET-EXTENSIONS in the SET command or LOG-APPLY in the COPY command.

CKZ452E **A dynamic allocation error occurred for the data set *data_set_name*: error code = *hexadecimal_number* hex, info code = *hexadecimal_number* hex**

Explanation:

The product encountered an error with the specified codes while trying to allocate the specified data set.

User response:

See the z/OS MVS Programming Authorized Assembler Services Guide for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ453E **You must specify a valid DD statement for *dd_name* DD name on the DD specification panel**

Explanation:

An invalid or an empty value for the *dd_name* DD name was entered.

User response:

Enter a valid value for the *dd_name* DD name on the **Db2 tablespace clone DD Specification** panel.

CKZ454W **Inconsistent member ID settings for data sharing group *group_name***

Explanation:

A member ID should be specified either for all members of the data sharing group or for none of them.

User response:

Ensure that the member ID is specified either for all members of the data sharing group or for none of them.

CKZ455W **Member IDs in data sharing group *group_name* are not unique**

Explanation:

All members of the data sharing group must have unique member IDs.

User response:

Make sure that each member of the data sharing group has a unique member ID.

CKZ456I Operation canceled by user

Explanation:

The CANCEL command was entered to cancel the last operation.

User response:

No action is required.

CKZ457W The log apply parameters were successfully updated for the selected profiles, except the ones that are displayed on this panel

Explanation

The log apply parameters were not updated for the profiles that are displayed on this panel. The list of reasons why this might happen are:

- DATA-MOVER PGM is set to SRCVSCP.Y.
- The profiles do not allow update access.
- The profiles are not at the current version.
- An internal error of the VSAM data repository routine occurred.

User response:

Check that the profiles that are displayed on this panel can be edited.

CKZ458E The log apply parameters cannot be updated for any of the selected profiles

Explanation

The log apply parameters were not updated for the profiles that are displayed on this panel. The list of reasons why this might happen are:

- DATA-MOVER PGM is set to SRCVSCP.Y.
- The profiles do not allow update access.
- The profiles are not at the current version.
- An internal error of VSAM data repository routine occurred.

User response:

Check that the profiles that are displayed on this panel can be edited.

CKZ459E No profiles are displayed

Explanation:

The primary command that was entered acts on the profiles that are listed on the panel, but no profiles are listed.

User response:

Modify the Profile Like or Creator Like fields to display the desired profiles and retry the command.

CKZ460E The DD name *dd_name* is duplicated in Product DD list and in User DD list

Explanation:

More than one DD with the same name is specified in the *job_type* Job DD Specification.

User response:

Change one of the DD names.

CKZ461E You must specify a valid LISTDEF statement before generating jobs

Explanation:

The STOGROUP parameter cannot be used when specifying CATALOG-PREFETCH.

User response:

Specify a valid LISTDEF statement and retry.

CKZ462E You must enter a valid Exclude mask for RENAME command before generating jobs

Explanation:

An invalid exclude mask was entered.

User response:

Enter a valid exclude mask and retry.

CKZ463E Invalid Value - Enter a valid DD name

Explanation:

An invalid DD name was entered.

User response:

Enter a valid DD name. A valid DD name is 1 through 8 alphanumeric or national (@, #, \$) characters. The first character must be alphabetic or national.

CKZ464E Invalid Value - You must enter "TS", "TP", "IX", "IP", "SG", or blank Apply To Type for "VCAT" Attribute Name

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ465E Empty list - No items were found in the list

Explanation:

The list cannot be empty.

User response:

Add at least one item to the list.

CKZ466E **The number of items must be less than or equal to *maximum_number***

Explanation:

This list has a limit on the maximum number of items.

User response:

Remove list items so that its size does not exceed the limit.

CKZ467E **Invalid Value - You must enter "TB", "SG", or blank Apply to Type for "KEYLABEL" Attribute Name**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ468E **You must enter the CATWORK-DSN MASK parameter in the Db2 subsystem COPY command because System Level Backup Type is DB2SLB and Use Dumptapes is YES**

Explanation:

The CATWORK-DSN MASK field is required if the cloning scenario uses a system level backup and the RESTORE-FROM-DUMPTAPES command is specified.

User response:

Enter a valid CATWORK-DSN MASK value in the COPY command.

CKZ469E **You must provide source and target ICF catalogs because System Level Backup Type is DB2SLB and Use Dumptapes is YES**

Explanation:

At least one pair of ICF catalogs is required if the cloning scenario uses a system level backup and the RESTORE-FROM-DUMPTAPES command is specified.

User response:

Enter at least one pair of source and target ICF catalogs.

CKZ470E **The VOLBKUP-DDN parameter is required when RENAME TYPE is set to SAFE**

Explanation:

A required parameter for the RENAME command was not specified. When RENAME TYPE is SAFE, VOLBKUP-DDN is required.

User response:

Specify a value for the VOLBKUP-DDN parameter and retry.

CKZ471E **The template variables are not selected**

Explanation:

The template variables were not selected. They must be selected when TEMPLATE-VARIABLE for the SET command is "YES."

User response:

Either select the template variables or change the value of TEMPLATE-VARIABLE field for the SET command.

CKZ472E **The attribute changes are not defined**

Explanation:

When INCLUDE DDL-ATTRIBUTE-CHANGE is set to "YES", the attribute changes on a separate panel must be defined.

User response:

Either use the D command to define the attribute changes or change the INCLUDE DDL-ATTRIBUTE-CHANGE field.

CKZ473E **The names and object types of the source and target objects are not specified**

Explanation:

When INCLUDE OBJECT-TRANSLATE is set to "YES", the names and object types on a separate panel must be specified.

User response:

Either use the O command to specify the names of the source objects and the new target object names in pairs or change the INCLUDE OBJECT-TRANSLATE field.

CKZ474E **The DD information for the templates is not defined**

Explanation:

When INCLUDE JOB-TEMPLATE is set to "YES", the DD information for the templates must be supplied.

User response:

Either use the J command to supply DD information for the templates or change the INCLUDE JOB-TEMPLATE field.

CKZ475E **Invalid Value - Enter a valid IP address**

Explanation:

The format of the specified IP address is incorrect.

User response:

Correct the specified IP address.

CKZ476E **Invalid Value - Enter a valid IPv4 address**

Explanation:

The format of the specified IPv4 address is incorrect.

User response:

Correct the specified IPv4 address.

CKZ477E **Invalid Value - Enter a valid IPv6 address**

Explanation:

The format of the specified IPv6 address is incorrect.

User response:

Correct the specified IPv6 address.

CKZ478E **Invalid Value - Enter a valid Exclude Mask**

Explanation:

An invalid Exclude Mask value was entered.

User response

Enter a valid Exclude Mask value. Syntax rules for exclude masks are as follows:

- A single asterisk (*) represents exactly one DSN qualifier of any value. For example: *. or *. or *
- A double asterisk (**) represents 0 to nn DSN qualifiers of any value. Double asterisks (**) cannot appear with any other characters within a qualifier. Three or more adjacent asterisks (***) are not allowed within a qualifier. Double asterisks (**) can be used more than once in a source rename mask.
- A percent sign (%) represents one non-blank character.
- An exclamation point (!) represents one national character: @ # \$.
- A less-than sign (<) represents one non-numeric character, national symbols included.
- A greater-than sign (>) represents one numeric character.

For more information on masks, refer to the [“Filtering pattern masks”](#) on page 26 topic.

CKZ479E **Invalid Value - You must enter a "BUFFERPOOL", "CLOSE", "DATACAPTURE", "GBPCACHE", "LOG", "PRIQTY", "SECQTY", "STOGROUP", "TRACKMOD", "VCAT", or "KEYLABEL"**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ480E **Invalid Value - Please enter a valid value for "GBPCACHE" Attribute**

Name: "CHANGED", "ALL", "SYSTEM", "NONE", or blank

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ481E **Invalid Value - Please enter a valid value for "GBPCACHE" Attribute Name: "CHANGED", "ALL", "NONE", or blank**

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ482E **Invalid Value - Enter a valid group attach name**

Explanation:

An invalid group attach name was entered.

User response:

Enter a valid group attach name. This name can be up to four alphanumeric or national (@, #, \$) characters long. The first character must be alphabetic or national.

CKZ483E **The group attach name was not defined for DB2 subsystem ssid**

Explanation:

The group attach name was not defined for the subsystem that is specified in the message, but it is required when "Use group attach name" is set to "YES" on the Source and Target DB2 subsystems panel.

User response:

Either change "Use group attach name" to "NO" or define a group attach name for the specified DB2 subsystem under Administrator Functions on the main menu.

CKZ484E **You cannot use the group attach name for non-data sharing DB2 subsystem: SSID = ssid**

Explanation:

A group attach name was specified for a non-data sharing DB2 subsystem.

User response:

Change the value of "Use group attach name" field to "NO" on the Source and Target DB2 subsystems panel.

CKZ485W **The number of members in the data sharing group *group_name* exceeds the maximum value**

Explanation:

The maximum number of members in a data sharing group is 32.

User response:

Remove some members from the data sharing group that is described in the message text.

CKZ486E **Members of the data sharing group *group_name* have different group attach names**

Explanation:

The group attach name of all members of the data sharing group must be the same.

User response:

Use Administrator Functions from the main menu to set up the same group attach name for each member of the data sharing group that is described in the message text.

CKZ487E **Invalid Value - Please enter a valid GMT offset**

Explanation:

An invalid GMT offset was specified.

User response:

Enter a valid GMT offset in the field. Valid formats are *+hh:mm* and *-hh:mm*.

CKZ488E **The specified GMT offset is out of range**

Explanation:

The valid range of GMT offset is from -12:00 to +14:00.

User response:

Change the value of GMT offset.

CKZ489I **No profiles are processed**

Explanation:

The action cannot be performed for selected profiles.

User response:

No action is required.

CKZ490E **REBUILD-UNMATCHED-TARGET-INDEXES cannot be enabled until both REBUILD-INDEXES-EXECUTE and PROCESS-UNMATCHED-TARGET-INDEXES are set to "YES"**

Explanation:

For REBUILD-UNMATCHED-TARGET-INDEXES(YES), REBUILD-INDEXES-EXECUTE(YES) and PROCESS-UNMATCHED-TARGET-INDEXES(YES) are required.

User response:

Set "YES" value for both REBUILD-INDEXES-EXECUTE and PROCESS-UNMATCHED-TARGET-INDEXES and then change the value for REBUILD-UNMATCHED-TARGET-INDEXES.

CKZ491E **The *parameter*(YES) is not valid when SKIP-LOG-APPLY is set to "YES"**

Explanation:

parameter(YES) is not allowed if the SKIP-LOG-APPLY keyword is set to "YES" in the LOG-APPLY command.

User response:

Change the value of either the SKIP-LOG-APPLY field or the *parameter* field.

CKZ492E **The *field_name* cannot be enabled when REBUILD-INDEXES-EXECUTE is not set to "YES"**

Explanation:

For *field_name* (YES), REBUILD-INDEXES-EXECUTE(YES) is required.

User response:

Set "YES" value for REBUILD-INDEXES-EXECUTE and then change the value for the field name that is listed in the message.

CKZ493I **You cannot use LOG-APPLY and UNLOAD-LOAD with USE-LAST-CONSISTENT-FLASHCOPY(Y). LOG-APPLY and UNLOAD-LOAD were disabled**

Explanation:

LA-ENABLE and UNLOAD-LOAD-ENABLE were set to "NO."

User response:

No action is required.

CKZ494E **The CHECK-INDEX-KEYS parameter must be enabled when *field_name* is set to YES**

Explanation:

A required parameter was not enabled.

User response:

Change the value of one of the parameters that is provided in the message.

CKZ495E **Invalid Value - Enter a valid Source Mask**

Explanation:

An invalid Source Mask value was entered.

User response

Enter a valid Source Mask value. Syntax rules for source masks are as follows:

- A single asterisk (*) represents 0 to nn characters of any value.
- A percent sign (%) represents one non-blank character.
- An exclamation point (!) represents one national character: @ # \$.
- A less-than sign (<) represents one non-numeric character, national symbols included.

- A greater-than sign (>) represents one numeric character.

For more information on masks, refer to the “[Filtering pattern masks](#)” on page 26 topic.

CKZ496E Invalid Value - Enter a valid Target Mask

Explanation:

An invalid Target Mask value was entered.

User response

Enter a valid Target Mask value for each source mask. Syntax rules for target masks are as follows:

- A single asterisk (*) represents 0 to *nn* characters of any value. The single asterisk (*) may only be used as the last character of the mask.
- A percent sign (%) represents one non-blank character.

For more information on masks, refer to the “[Filtering pattern masks](#)” on page 26 topic.

CKZ497E The INCLUDE JOB-TEMPLATE parameter must be set to "YES" if the CMDDDDNAME parameter is specified

Explanation:

A required parameter for the COPY command was not enabled.

User response:

Either clear the CMDDDDNAME field or change the value of INCLUDE JOB-TEMPLATE field.

CKZ498E The DD name specified in the CMDDDDNAME does not match any output DD name in the job template list

Explanation:

The CMDDDDNAME is the name of the output DD that will contain the job built using the JOB-TEMPLATE parameter of the COPY command.

User response:

On the COPY command panel, either clear the CMDDDDNAME field or use the J command to supply DD information for templates.

CKZ499E The READ-FROM-SERVER-CACHE parameter must be enabled when ENABLE-TARGET-PREFETCH is set to YES

Explanation:

A required parameter was not enabled.

User response:

Change the value of either the READ-FROM-SERVER-CACHE field or the ENABLE-TARGET-PREFETCH field.

CKZ500I The value of Remote connection type will be changed from DDF to CAF | TCPIP automatically

Explanation:

The DDF value that was entered in the **Remote connection type** field is obsolete. The value will be automatically changed to CAF or TCPIP.

User response:

No action is required.

CKZ501I The value of SKIP-LOG-APPLY will be changed from YES to NO automatically

Explanation:

A value of YES for SKIP-LOG-APPLY is not supported with DATA-MOVER PGM ADRDSSU, EMCAPI, or NONE. SKIP-LOG-APPLY will be changed to NO.

User response:

No action is required.

CKZ502I The value of SKIP-LOG-APPLY will be changed from NO to YES automatically

Explanation:

A value of NO for SKIP-LOG-APPLY is not supported. SKIP-LOG-APPLY will be changed to YES.

User response:

No action is required.

CKZ503I The value of SPACES-PER-LOG-APPLY-CALL will be set to blank automatically

Explanation:

The SPACES-PER-LOG-APPLY-CALL parameter is only valid if DATA-MOVER PGM(ADRDSSU), DATA-MOVER PGM(EMCAPI), or DATA-MOVER PGM(NONE) is specified for cross-LPAR cloning.

User response:

No action is required.

CKZ510I No objects match the filter

Explanation:

The list of objects is empty. There are no objects that meet the filter criteria.

User response:

Change the filter values or add new objects to the list.

CKZ511I Operation completed successfully

Explanation:

This message indicates that the operation has completed without error.

User response:

No action is required.

| | | |
|-----------------------|---|--|
| CKZ512E | The profile with given name already exists | Enter a valid member name in the field. |
| Explanation: | The name that you provided for the profile already exists. | |
| User response: | Enter a different unique profile name. | |
| CKZ513I | No item(s) selected | CKZ519E Data set is not partitioned |
| Explanation: | No item(s) were selected on selection panel. | Explanation: The specified data set already exists, but it is not partitioned. |
| User response: | Select one or more items to continue. | User response: Enter the name of either a partitioned data set or a non-existent data set. |
| CKZ514E | You do not have access rights to this profile | CKZ520E Cannot open file filename |
| Explanation: | An attempt was made to edit a profile that was created with no access rights by another user ID. Access to the profile is denied. | Explanation: A file cannot be opened. |
| User response: | Select a different profile to edit. | User response: If unable to determine the reason that the file open failed, contact IBM Software Support. Have available the listing that contains this message. |
| CKZ515E | You only have rights to view this profile | CKZ521E Cannot get text from skeleton |
| Explanation: | An attempt was made to edit a profile that was created with view only rights by another user ID. Editing the profile is not allowed. | Explanation: Text could not be read from a skeleton. |
| User response: | Use the V line command to view the profile. | User response: If unable to determine the reason that text cannot be retrieved from skeleton, contact IBM Software Support. Have available the listing that contains this message. |
| CKZ516E | Invalid Value - Please enter a Db2 subsystem ID | CKZ522I You have already chosen all available object mismatch types |
| Explanation: | You must specify a Db2 subsystem ID. You can enter a question mark (?) in the field to open a list of existing subsystems from which to choose. | Explanation: All available object mismatch types have already been selected. |
| User response: | Choose or enter a valid Db2 subsystem ID. | User response: No action is required. |
| CKZ517E | Invalid Value - Please enter a data set name | CKZ523E A dynamic allocation error occurred for the data set <i>data_set_name</i>: error code = <i>hexadecimal_number</i> hex, info code = <i>hexadecimal_number</i> hex |
| Explanation: | An invalid data set name was entered. | Explanation: The product encountered an error with the specified codes while trying to allocate the specified data set. |
| User response: | Enter a valid data set name in the field. | User response: See the z/OS MVS Programming Authorized Assembler Services Guide for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message. |
| CKZ518E | Invalid Value - Please enter a member name | CKZ524I Operation canceled by user |
| Explanation: | An invalid value was entered. | Explanation: The CANCEL command was entered to cancel the last operation. |
| User response: | | User response: |

No action is required.

CKZ525E Invalid Value - Please enter a valid control HLQ

Explanation:

An invalid high level qualifier was entered.

User response:

Enter a valid high level qualifier in the field.

CKZ526E Connection program load modules DSNALI, DSNHLI2, DSNWLI2, DSNTIAR, DSNHDECP were not found in specified load libraries for Db2 subsystem

Explanation:

The listed load modules were not found in the specified load libraries. The load library usually consists of a subsystem-specific DSNEXIT library, and the base DSNEXIT library and base DSNLOAD library for the current Db2 version.

User response:

Specify the data set that comprises the current load library concatenation for Db2 and that is used during batch job processing. To do this, use the **Db2 Subsystems** panel and line command E (Edit).

CKZ527E The number of items in the list exceeds the maximum value limit

Explanation:

The list has a limit on the maximum number of items.

User response:

Remove some items from the list.

CKZ528E The Db2 subsystem with given name already exists

Explanation:

The specified Db2 subsystem is already defined in the program.

User response:

Enter another Db2 subsystem name.

CKZ529I Db2 subsystem ssid was successfully selected

Explanation:

The specified Db2 subsystem was successfully selected.

User response:

No action is required.

CKZ530I The specified Db2 subsystem is not defined. Please provide the information for it

Explanation:

The specified Db2 subsystem could not be found in the Db2 control data set that is specified in the CLIST.

User response:

Fill in the required fields and save the Db2 subsystem.

CKZ531E Invalid Value - Enter a valid group attach name

Explanation:

An invalid group attach name was entered.

User response:

Enter a valid group attach name. This name can be up to four alphanumeric or national (@, #, \$) characters long. The first character must be alphabetic or national.

CKZ532W The same Db2 subsystem is specified in the Source SSID and the Target SSID, but object translation is not specified

Explanation:

The object translation feature allows you to copy table spaces and index spaces to the same Db2 subsystem with different names.

User response:

Make sure that the object translation feature is not required in this scenario, or enable the **Use object translation** parameter and provide object translation information.

CKZ533E Update is only valid for wildcard selections

Explanation:

Wildcards can only be edited in the object list.

User response:

Use the U (Update) line command for wildcards.

CKZ534E You must first specify an INCLUDE clause

Explanation:

LISTDEF must begin with an INCLUDE clause.

User response:

Specify an INCLUDE clause first. You can use M (Move) line command for this purpose.

CKZ535E Invalid Value - You must enter a "YES" or "NO"

Explanation:

An invalid value was entered.

User response:

Enter a valid value as described in the message text.

CKZ536E Invalid Value - Enter a valid DD name

Explanation:

An invalid DD name was entered.

User response:

Enter a valid DD name. The valid DD name is 1 through 8 alphanumeric or national (@, #, \$) characters. The first character must be alphabetic or national.

CKZ537E LISTDEF utility does not support all-inclusive lists

Explanation

You cannot use all-inclusive lists when the Wildcard field is set to YES and you define pattern-matching object names by using the following pattern-matching characters:

- '%' and '*' for database and storage group object types
- '%.%', '%.*', '*.%', '*.*' for other object types

User response:

Correct the pattern strings.

CKZ538I All items are already selected for the specified filtering criteria

Explanation:

There are no new items that can be selected.

User response:

Change the filter criteria.

CKZ539E The Db2 subsystem ssid is not defined

Explanation:

The Db2 subsystem that is specified on the **Primary Option Menu** panel could not be found in the Db2 control data set that is specified in the CLIST.

User response:

Either create the Db2 subsystem with the specified SSID or use a name of existing Db2 subsystem.

CKZ540E The Db2 subsystem ssid is not defined

Explanation:

The specified Db2 subsystem could not be found in the Db2 control data set that is specified in the CLIST.

User response:

Either create the Db2 subsystem with the specified SSID or use a name of existing Db2 subsystem.

CKZ541E Location is not specified for the Db2 subsystem ssid

Explanation:

The required parameter was not specified.

User response:

Edit the specified Db2 subsystem and retry.

CKZ542E Db2 object not found in Db2 catalog tables

Explanation:

A problem occurred while accessing information in the Db2 catalog tables. The Db2 object may have been dropped since the last time the object list was retrieved.

User response:

Refresh the list of objects to see all available objects.

CKZ543E Required parameter is not defined

Explanation:

A required parameter was not specified.

User response:

Specify the required parameter and retry.

CKZ544E Invalid value for attribute_name attribute

Explanation:

The specified attribute name has incorrect values for some fields.

User response:

Refer to the panel help for a list of possible values.

CKZ545E Empty list - No items were found in the list

Explanation:

The list cannot be empty.

User response:

Add at least one item to the list.

CKZ546W The combination of field values Source availability during cloning = YES and Target consistency = NO may cause data inconsistency on target

Explanation:

If the table spaces and index spaces are in read-write status, specifying source availability during cloning will not guarantee consistent data on the target without also using target consistency, as there might be uncommitted updates to the source data.

User response:

Make sure the field values are correct.

CKZ547I The UNLOAD-LOAD feature will be disabled

Explanation:

If target consistency is enabled, then the UNLOAD-LOAD feature is automatically switched off.

User response:

No action is required.

CKZ548E Invalid Value - The specified GMT offset is out of range

Explanation:

The valid range of the GMT offset is from -12:00 to +14:00.

User response:

Change the value of the GMT offset.

CKZ549E Invalid Value - Enter a valid Db2 plan name

Explanation:

An invalid Db2 plan name was entered.

User response:

Enter a Db2 plan name. This name can be up to eight characters long, each of which is an alphanumeric, a national (@, #, \$), or the underscore (_) character. The first character must be alphabetic or national.

CKZ550E Cloning profile not found in VSAM data repository

Explanation:

A problem occurred while accessing information in the VSAM data repository. The cloning profile may have been deleted since the last time the profile list was retrieved.

User response:

Refresh the list of profiles to see all available profiles.

CKZ551E You must specify a valid DD statement for *dd_name* DD name on the DD specification panel

Explanation:

The valid DD statement must contain the DSN parameter in the format *DSN=data-set-name(member-name)*.

User response:

Enter a valid value for the specified DD name on the **DD Specification** panel.

CKZ556E Properties line command is not valid for wildcard selection

Explanation:

Viewing properties is not valid for wildcard selections. If the Wildcard field is set to YES, the substitution of wildcards will be used according to the specified filter criteria in the source job at runtime.

User response:

Clear the line command from the Cmd line.

CKZ557E Insufficient region size. Available region size of at least 50000 is required

Explanation:

The available region size is insufficient to run the Db2 Cloning Tool ISPF interface.

User response:

Contact your system administrator to increase the region size.

CKZ563E You must specify Minilog data set HLQ if Target consistency is set to YES

Explanation:

The **Minilog data set HLQ** field is blank, and **Target consistency** is set to YES.

User response:

Enter the **Minilog data set HLQ** by using the **Minilog Data Set Attributes** panel.

CKZ564E You must specify RTS data set HLQ if Clone statistics is set to YES

Explanation:

The **RTS data set HLQ** field is blank and **Clone statistics** is set to YES.

User response:

Enter the **RTS data set HLQ** by using the **Statistics Copy Options** panel.

CKZ565E Use data masking | Target job restartability cannot be enabled when Process extensions in parallel is set to YES

Explanation:

The value YES is not valid for **Use data masking** or **Target job restartability** if the **Process extensions in parallel** keyword is set to YES.

User response:

Change the value of either the **Use data masking** or **Target job restartability** field, or change the value in the **Process extensions in parallel** field.

CKZ566E Process extensions in parallel cannot be enabled when Use data masking | Target job restartability is set to YES

Explanation:

The value YES is not valid for **Process extensions in parallel** if the **Use data masking** or **Target job restartability** field is set to YES.

User response:

Change the value of either the **Process extensions in parallel** field or the **Use data masking** or **Target job restartability** field.

CKZ567E The target data set VCAT must be provided when Copy if no target object is set to YES

Explanation:

The target data set VCAT was not specified. It is required when **Copy if no target object** is set to YES.

User response

Either specify the target data set VCAT or change the value of the **Copy if no target object** field on the **Advanced Cloning Options** panel for the cloning profile. To specify the target data set VCAT, use one of the following:

- The object translate VCAT command on the **Target Options** panel for the cloning profile.
- The default VCAT on the **Advanced Cloning Options** panel for the cloning profile.
- The target VCAT for the target Db2 SSID.

CKZ568E **The Use job templates parameter must be set to YES if Use command job template is set to YES**

Explanation:

The value YES is required for the **Use job templates** parameter when **Use command job template** is set to YES.

User response:

Either set **Use job templates** to YES or set **Use command job template** to NO.

CKZ569E **Input template data set and Input template member must be specified if Use command job template is set to YES**

Explanation:

Input template data set or **Input template member** was not specified. Both are required when **Use command job template** is set to YES.

User response:

Specify both **Input template data set** and **Input template member**.

CKZ700E **An error occurred while opening the Db2 load libraries: RC = return_code**

Explanation:

The product encountered an error with the specified return code while opening the Db2 load library data sets.

User response:

Make sure that the load library data sets that are specified on the **Db2 Subsystem Parameters** panel exist, and that you have the proper authority to read them.

CKZ701E **An error occurred while attaching the Db2 attachment facility subtask: RC = return_code**

Explanation:

The product encountered an error with the specified return code while attaching the Db2 attachment facility subtask.

User response:

See the Db2 for z/OS codes information for details about the return code.

CKZ702E **The task is not running APF authorized**

Explanation:

The task requires load module &PRD.XDBT to be APF authorized.

User response:

Set up APF authorization for load module &PRD.XDBT.

CKZ703S **The Db2 attachment facility subtask ended unexpectedly: RC = return_code**

Explanation:

The product encountered the error with the specified return code.

User response:

Make sure that the Db2 subsystem is active.

CKZ704E **The specified user ID user_id and password are invalid**

Explanation:

A valid user ID and password are required to establish a connection to the Db2 subsystem.

User response:

Specify a valid user ID and password.

CKZ705E **The specified password for user ID user_id has expired**

Explanation:

A valid user ID and password are required to establish a connection to the Db2 subsystem.

User response:

Contact your system administrator for a valid password.

CKZ706E **Access for the specified user ID user_id has been revoked**

Explanation:

A valid user ID and password with proper authority are required to establish a connection to the Db2 subsystem.

User response:

Make sure that you have the proper authority to connect to the Db2 subsystem.

CKZ707E **An error occurred while performing authentication: SAF RC = return_code, RC = return_code, RSN = reason_code**

Explanation:

You must have the proper authority to access the Db2 subsystem.

User response:

Make sure that you have the proper authority. See the z/OS Security Server RACF Callable Services guide for information about the codes.

CKZ708E **An invalid dynamic allocation parameter was specified: code = code**

Explanation:

The DD allocation for the Db2 load library data set failed.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ709E **A dynamic allocation error occurred: info code = info_code, error code = error_code**

Explanation:

The product encountered an error with the specified codes.

User response:

See the z/OS MVS Programming Authorized Assembler Services Guide for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ710E **Dynamic allocation query error occurred: info code = info_code, error code = error_code**

Explanation:

The product encountered an error with the specified codes.

User response:

See the z/OS MVS Programming Authorized Assembler Services Guide for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ711E **Dynamic free error occurred: info code = info_code, error code = error_code**

Explanation:

The product encountered an error with the specified codes.

User response:

See the z/OS MVS Programming Authorized Assembler Services Guide for information about the codes. If unable to determine the cause of the error, contact

IBM Software Support. Have available the listing that contains this message.

CKZ712E **Dynamic concatenation error occurred: info code = info_code, error code = error_code**

Explanation:

The product encountered an error with the specified codes.

User response:

See the z/OS MVS Programming Authorized Assembler Services Guide for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ713E **SQL error occurred: SQL code = code, SQL state = code**

Explanation:

The product encountered an error with the specified codes.

User response:

See Db2 for z/OS codes for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ714E **SQL error_text (SQL code: code. Program: program_name. Statement: line_number. Type: type)**

Explanation:

The product encountered an error with the specified code.

User response:

See Db2 for z/OS messages for information about the code. If you are unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ715S **Db2 attachment facility error occurred: function = code, RC = return_code, reason = reason_code**

Explanation:

The product encountered an error with the specified codes.

User response:

See the Db2 for z/OS Application Programming and SQL Guide for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ716E **Input Db2 command is too long**

Explanation:

The Db2 command failed because the command is not valid.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ717E **Error occurred while making an IFI COMMAND call:**
ifi_result_messages IFCARC1 =
ifi_return_code, IFCARC2 =
ifi_reason_code

Explanation:

The product encountered an error while making the Db2 request that is described in the message.

User response:

See the Db2 for z/OS codes for information about the codes. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ801E **Pgm: *program name* Stmt:**
statement Type: type

Explanation:

This message is used to convert SQL return code information into a text message. The data from the SQLCA is called using DSNTIAR and formatted into this message.

User response:

Ensure that the qualifier value in the Run Table Space Cloning package binds job matches the creator name used in the Create Table Space Cloning log apply table job. Refer to the DB2 for z/OS: SQL Reference for your version of DB2 to resolve.

CKZ810E **Invalid CNUM parm. Valid parms are ON, OFF, or blank**

Explanation:

CNUM was issued with an invalid parameter. Issuing CNUM with no parameter acts as an ON/OFF toggle. ON and OFF are the only parameters accepted. ON turns the CNUM display on. OFF turns the CNUM display off.

User response:

Use a valid CNUM parameter (ON, OFF, or blank)

CKZ811E **Invalid COLS parm. Valid parms are ON, OFF, or blank**

Explanation:

COLS was issued with an invalid parameter. Issuing COLS with no parameters acts as an ON/OFF toggle. ON and OFF are the only parameters accepted.

User response:

Enter COLS ON or COLS OFF. COLS ON turns the COLS display on; COLS OFF turns the COLS display off.

CKZ812I **The FIND command requires a match string**

Explanation:

No parameters were specified with the FIND command. Specify a string to match.

User response:

Enter FIND parameters.

CKZ813E **The RFIND key works only after a FIND character string is entered**

Explanation:

A repeat FIND (RFIND) was issued before a FIND command was issued. You must issue FIND before RFIND will work.

User response:

Issue FIND prior to attempting to issue RFIND.

CKZ814E **An unknown column *column name* was specified**

Explanation:

The column you specified with the SORT command is not known.

User response:

Verify that you correctly typed the name of the column or select another column.

CKZ815E **SORT is not supported for the specified column**

Explanation:

The column you attempted to SORT is not supported as a column on which to sort.

User response:

Refer to the sort columns listed on the Define Sort Columns panel for a list of valid columns on which the sort can be based and redefine the sort.

CKZ816E **Max Sort Columns exceeded. Sorting first 9 columns**

Explanation:

More columns were selected for sorting than are supported. Nine columns can be selected. Under certain circumstances the limit is less than nine, due to internal constraints. For example, sorting a date field can be implemented by three sorts of partial column fields. In that case, the column would

User response:

Specify the appropriate allowable maximum number of sort columns.

CKZ817E **Invalid column selection. Set cursor to valid column**

Explanation:

An invalid column was selected.

User response:

Set the cursor to a valid column.

CKZ818E Invalid command parameters

Explanation:

Invalid command parameters were entered.

User response:

Correct the command input and resubmit.

**CKZ819E Invalid place for moved column.
Cannot move source column to the
new position**

Explanation:

The source column cannot be moved to the new position.

User response:

Correct the command input and resubmit.

**CKZ820E Not enough space for scrolling
unfixed columns**

Explanation:

Not enough space is available for scrolling unfixed columns

User response:

Shrink the fixed area.

**CKZ821E Operation not valid for specified
column**

Explanation:

An invalid operation was entered.

User response:

Enter a valid operation.

CKZ822E Fixed columns cannot be hidden

Explanation:

An attempt was made to hide a fixed column, but fixed columns cannot be hidden.

User response:

Either make a selected column unfixed or select another column to hide.

**CKZ823E Invalid value entered for column
size: non-numeric data**

Explanation:

An invalid Cmd value was entered. This must be a number between the values in the MIN and MAX fields.

User response:

Either remove the invalid number or enter a valid one.

**CKZ824E Invalid value entered for column
size: out of range**

Explanation:

An invalid Cmd value was entered. This must be a number between the values in the MIN and MAX fields. MIN is the smallest acceptable value. MAX is the largest acceptable value.

User response:

Either remove the invalid number or enter a valid one.

**CKZ825E SIZE is not supported for the
specified column**

Explanation:

SIZE is not supported for the specified column. You cannot change the size of the column.

User response:

You can change the size of another column, in which the minimum and maximum sizes are unequal.

**CKZ940E Invalid selection character. "F"
and "U" are valid**

Explanation:

An invalid Cmd character was entered. Valid characters are F (FIX) and U (UNFIX).

User response:

Either remove the invalid character or enter a valid one.

**CKZ941E Column move failed - invalid
location**

Explanation:

An attempt to move a column was made, but the attempt failed because the new location was invalid. The new column number cannot be greater than the number of columns.

User response:

Specify a column number that is not greater than the number of columns.

**CKZ942E Invalid column size. Column size
must be numeric**

Explanation:

An invalid Cmd value was entered. Column size must be a number between the values in the MIN and MAX fields.

User response:

Either remove the invalid number or enter a valid one.

**CKZ943E Invalid column size. The specified
value is out of range**

Explanation:

An invalid Cmd value was entered. Column size must be a number between the values in the MIN and MAX fields. MIN is the smallest acceptable value. MAX is the largest acceptable value.

User response:

Either remove the invalid number or enter a valid one.

**CKZ944E Total fixed column sizes cannot
exceed screen size**

Explanation:

The Cmd values that are entered would have resulted in the sum of the FIXed column sizes to exceed the screen size. Because FIXed columns are always displayed, they must fit on the screen. The FIXed columns contain an F or P in the Fix column.

User response:

Either change the FIXed column sizes so that the total is less than the screen size, or cancel to return to the previous panel.

CKZ945E Configuration request failed - at least one unfixed column would not be displayed

Explanation:

The requested column sizes would cause at least one unfixed column to become undisplayable. The cursor is positioned on the value where the problem was detected. The unfixed area on the screen would be too small to show the column where the cursor is placed.

User response

To correct the problem, do one of the following:

- Decrease the size of the column at which the cursor is pointing so that it can fit in the available unfixed area.
- Set the unfixed area to its maximum size (width).
- Decrease the size of the fixed area.
- Cancel to return to the previous panel.

CKZ946E Configuration request failed - the unfixed area would be too small to display this column

Explanation:

The requested column sizes would make the UNFIXed column at which the cursor is positioned undisplayable. The UNFIXed area on the screen would be too small to show this column.

User response:

You can shrink the FIXed area by either unfixing columns or making FIXed columns smaller.

CKZ947E Configuration request failed - not all columns can be displayed

Explanation:

Fixing the requested columns would shrink the available area for unfixed columns so that some may not display. The cursor is placed on a row that represents one of the columns that would cause the error if FIXed.

User response:

To change column sizes, cancel out of the CFIX function and invoke the CSIZE function. Either cancel to exit CFIX with no change, or blank out one or more FIX selections until an allowable fixed size is reached.

CKZ948E Invalid FIXed selections. Operation would not leave enough space for this column

Explanation:

Fixing the columns requested would make at least one unfixed column undisplayable. The cursor is positioned on the row that represents one of the unfixed columns that would cause an error where the minimum displayable size would not fit in the available screen area.

User response

Shrink the requested fixed area by doing one of the following:

- Request fewer fixed columns.
- Unfix one or more fixed columns.
- Cancel out of CFIX, then invoke CSIZE to shrink one or more fixed columns enough so that all unfixed columns have the space they require.

CKZ949E Duplicate Cmd values entered

Explanation:

Duplicate Cmd numbers were entered. The cursor points to the second instance of a Cmd value.

User response:

Either change this value, clear it, or exit the CORDER function.

CKZ950E Invalid sort number. Enter a valid numeric digit

Explanation:

An invalid character was entered in the Srt column.

User response:

Enter a valid character. Valid characters include the digits 1-9, or the number of sortable columns, whichever is less.

CKZ951E Duplicate sort sequence number

Explanation:

The same sort sequence number was entered for more than one column. The screen is positioned to the second instance.

User response:

Enter a unique sort sequence number.

CKZ952E Sort sequence skips a number

Explanation:

The selected sorting sequence skips a number. The screen is positioned to the selection after the missing number in the sequence.

User response:

You must specify a valid sort sequence that does not skip a number.

CKZ953E Invalid Dir entered. Direction must be A (ascending) or D (descending)

Explanation:

The selected sorting direction is invalid.

User response:

Enter a valid value. Valid values include "A" for ascending, "D" for descending or leave the field blank. If left blank, the default (ascending) is used.

CKZ954E Dir not valid without Ord

Explanation:

A sorting direction (Dir) was selected for a column that was not selected to be sorted (Ord). Sorting direction is only a valid choice for selected columns.

User response:

You must specify a column to be sorted (Ord) before specifying a sort order direction.

CKZ955E Fixed columns cannot exceed screen size

Explanation:

More columns were selected to be FIXEd than will fit on the screen.

User response:

Remove the FIXEd (F) selection character from one or more columns.

CKZ956E Invalid entry. Cmd values must be numeric

Explanation:

An invalid Cmd value was entered. Cmd values must be numeric.

User response:

Either remove the invalid value or enter a valid numeric.

CKZ957E Invalid entry for permanent column

Explanation:

An invalid entry was made for a permanent column.

User response:

Enter a valid value.

CKZ958E Invalid entry for fixed column

Explanation:

An invalid Cmd value was entered for a FIXEd column. Valid selections for a FIXEd column are numeric values from 1-*n*, where *n* is the total number of fixed columns.

User response:

Either remove the invalid number or enter a valid number.

CKZ959E Invalid entry for unfixed column

Explanation:

An invalid Cmd value was entered for an UNFIXEd column. The number must be less than the total number of columns, and greater than the number of FIXEd columns.

User response:

Either remove the invalid number or enter a valid number.

CKZ960E Invalid Column Function value. Valid values: 1, 2, 3, 4 and 5

Explanation:

An invalid character was entered in the Column Function field. Valid characters are 1, 2, 3, 4, and 5.

User response:

Correct the field or issue the CANCEL command.

CKZ961E Invalid Permanent View value. Valid values: Y, N

Explanation:

An invalid character was entered in the Permanent View field. Valid characters are Y and N.

User response:

Correct the field or issue the CANCEL command.

CKZ962E Invalid Reset View value. Valid values are Y, N

Explanation:

An invalid character was entered in the Reset View field. Valid characters are Y (Yes), or N (No).

User response:

Correct the field or issue the CANCEL command.

CKZ963E Invalid Stop Sorting value. Valid values: Y, N

Explanation:

An invalid character was entered in the Stop Sorting field. Valid characters are Y (yes), or N (no).

User response:

Correct the field or issue the CANCEL command.

CKZ964E Invalid data set name

Explanation:

The data set name entered is syntactically incorrect. A data set name can be one name segment, or a series of joined name segments. Segments are limited to eight characters, the first of which must be alphabetic (A to Z) or special (# @ \$). The remaining seven characters are either alphabetic, numeric (0 - 9), special, a hyphen (-). Name segments are separated by a period (.). Including all name segments and periods, the length of the data set name must not exceed 44 characters. Thus, a maximum of 22 name segments can make up a data set name.

User response:

Enter a valid data set name.

CKZ965E Invalid member name

Explanation:

A member name can be up to eight characters long, and it can consist of the characters A-Z, 0-9, \$, #, and @.

User response:

Enter a valid member name.

CKZ966E Unable to allocate the report file

Explanation:

An error occurred when allocating the data set for the report.

User response:

If unable to determine the reason for the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ967E Unable to open the report file

Explanation:

An error occurred when opening the data set for the report.

User response:

If unable to determine the reason for the error, contact IBM Software Support. Have available the listing that contains this message.

CKZ968E Invalid selection character. "H" and "U" are valid

Explanation:

An invalid Cmd character was entered. Valid characters are H (Hide) and U (Unhide).

User response:

Either remove the invalid character or enter a valid one.

CKZ970E TBCREATE failed. RC=return_code

Explanation:

The TBCREATE command was issued to create a VIEW, but it failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to the ISPF Services Guide under TBCREATE.

CKZ971E TBOPEN failed. RC= return_code

Explanation:

The TBOPEN command was issued to open an existing VIEW, but the command failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to the ISPF Services Guide under TBOPEN.

CKZ972E TBCLOSE failed. RC= return_code

Explanation:

The TBCLOSE command failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to the ISPF Services Guide under TBCLOSE.

CKZ973E TBDELETE failed. RC= return_code

Explanation:

The TBDELETE command failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to the ISPF Services Guide under TBDELETE.

CKZ974E TBMOD failed. RC= return_code

Explanation:

The TBMOD command failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to the ISPF Services Guide under TBMOD.

CKZ975E TBGET failed. RC= return_code

Explanation:

The TBGET command failed with a (hex) return code as indicated in the message.

User response:

Review ISPTLIB allocation and data set characteristics. Review security controlled access to ISPTLIB data sets. For information about return codes, refer to the ISPF Services Guide under TBGET.

CKZ976E View Table is in use

Explanation:

A view input library is in use. Views are read from the ISPTLIB data set(s) and saved to the ISPTABL data set. To save and use report customizations that are created via the CSET command, ISPTABL and ISPTLIB must be allocated. The TBSTATS service has indicated that ISPTLIB is not allocated. Only temporary views will be available.

User response:

Issue the CSET command and access online help for details on the online report customization options available through the CSET command. In addition, review the ISPTLIB and ISPTABLE allocations. For information about ISPTLIB and ISPTABL, refer to the ISPF user guides for your version of ISPF. Refer to the configuration chapter of this user guide for the recommended method of allocating ISPTLIB and ISPTABL.

CKZ977E **View Library not allocated**

Explanation:

A view input library has not been allocated. Views are read from the ISPTLIB data set(s) and saved to the ISPTABL data set. To save and use report customizations that are created via the CSET command, ISPTABL and ISPTLIB must be allocated. The TBSTATS service has indicated that ISPTLIB is not allocated. Only temporary views will be available.

User response:

Issue the CSET command and access online help for details on the online report customization options available through the CSET command. In addition, review the ISPTLIB and ISPTABLE allocations. For information about ISPTLIB and ISPTABL, refer to the ISPF user guides for your version of ISPF. Refer to the configuration chapter of this user guide for the recommended method of allocating ISPTLIB and ISPTABL.

CKZ00001I *message_text*

Explanation:

This message is informational.

User response:

No action is required.

CKZ00002E **SUBTASK *subtask_number*, CLOSE
ACB ERROR, R15=*return_code*,
ERFLG=*ACBERFLG***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00003I **SUBTASK *subtask_number*, DB2
LDS CLOSE OK, DDN=*ddname*,
DSN=*data_set_name***

Explanation:

This message is informational.

User response:

No action is required.

CKZ00005I **SUBTASK *subtask_number*, DB2
LDS ALLOCATION OK,
*data_set_name***

Explanation:

This message is informational. LDS is a VSAM linear data set.

User response:

No action is required.

CKZ00006E **Internal error: bad buffer length.
Info: *diagnostic_info1*,
diagnostic_info2, *diagnostic_info3***

Explanation:

This message indicates an internal programming error.

User response:

Contact IBM Software Support.

CKZ00007I **SUBTASK *subtask_number*, DB2
LDS FREE OK, *data_set_name***

Explanation:

This message is informational. LDS is a VSAM linear data set.

User response:

No action is required.

CKZ00009E **SUBTASK *subtask_number*,
GENCB ACB ERROR,
R15=*return_code*, R0=*reason_code***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00010E **SUBTASK *subtask_number*,
GENCB EXL ERROR,
R15=*return_code*, R0=*reason_code***

Explanation:

This is an internal error. EXL is the VSAM exit list.

User response:

Contact IBM Software Support.

CKZ00011E **SUBTASK *subtask_number*,
GENCB RPL ERROR,
R15=*return_code*, R0=*reason_code***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00012E **INTERNAL ERROR,
RC=*return_code*, RS=*reason_code*,
PLEASE CONTACT SUPPORT
CENTER**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00013E Subtask *subtask_number*, data masking page error, RC=*return_code*, RS=*reason_code*, DSN=*data_set_name*, page#=*page_number_in_data_set*

Explanation:

The data masking page processor returned an error.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00014E Subtask *subtask_number*, data masking DSN error, RC=*return_code*, RS=*reason_code*, DSN=*data_set_name*

Explanation:

Data masking data set initialization returned an error.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00015I Subtask *subtask_number*, data masking pages: processed = *number_of_pages_processed*, changed = *number_of_pages_changed*

Explanation:

This message is informational.

User response:

No action is required.

CKZ00018I ?? SEGMENT HEADER BITS, FIRST BLOCK, ASSUMING LOB TS

Explanation:

This is an informational message.

User response:

No action is required.

CKZ00019I LOB TS SEGMENT HEADER BITS

Explanation:

This is an informational message.

User response:

No action is required.

CKZ00020E LOCK ALLOCATE ERROR, RS=*reason_code*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00021E LOCK RELEASE ERROR, RS=*reason_code*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00022E SUBTASK *subtask_number*, MODCB ERROR, R15=*return_code*, R0=*reason_code*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00023I Subtask *subtask_number*, Log Apply Page Changes = *decimal_number_of_pages*

Explanation:

Page with log records applied.

User response:

No action is required.

CKZ00024E SUBTASK *subtask_number*, OPEN ACB ERROR, R15=*return_code*, ERFLG=*ACBERFLG*

Explanation:

This may or may not be an internal error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00025I SUBTASK *subtask_number*, DB2 LDS OPEN OK, HI ALLOC RBA=*vsam_rba*, HI USED RBA=*vsam_rba* DDN=*ddname* DSN=*data_set_name*

Explanation:

This message is informational.

User response:

No action is required.

CKZ00026E Unable to load program: *load_module_name*

Explanation:

A log apply module could not be loaded.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00027E Bad return from *load_module_name*, RC=*return_code*, RS=*reason_code*

Explanation:

A log apply module could not be loaded.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00028E **Bad eyecatcher, B call,
first_8_characters_of_eyecatcher**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00029I **Log apply initialization successful**

Explanation:

Log apply is ready to begin processing log records.

User response:

No action is required.

CKZ00030E **Log apply initialization failure,
RC=return_code**

Explanation:

Log apply has failed to initialize. Check for log apply error messages preceding this message. See CKZ00031I for message output.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00031I **CKZG-B, message_text**

Explanation:

A message from log apply initialization call. It could be informational, a warning or an error.

User response:

No action is required, unless the call ends in an error.

CKZ00033W **PAGE SET: READ
ERRORS=decimal_number_of_errors,
WRITE
ERRORS=decimal_number_of_errors**

Explanation:

One or more errors were encountered while accessing the page set data sets in the target job.

User response:

Treat these errors as if they are DB2 IO errors on the target spaces. Contact your systems programmer.

CKZ00034I **Subtask subtask_number, Page
Set: PGRK=count_of_reads,
PGWK=count_of_writes,
PTT=count_of_points,
PGRW=count_of_writes,
PTBK=points_back_to_read_next_
page (V7)**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00035I **Subtask subtask_number,
inconsistent bit found in page,
DSN=data_set_name,
page=page_number_in_hex**

Explanation:

This message is a warning and may indicate a problem with page data.

User response:

Determine if the page data is correct. If not, rerun the source and the target jobs to reprocess the data set.

CKZ00036I **Subtask subtask_number, module,
allocation OK,
allocation_command**

Explanation:

This message is informational. The DD was successfully allocated using the allocation command that is listed in the message.

User response:

No action is required.

CKZ00037I **Subtask subtask_number, module,
free OK, free_command**

Explanation:

This message is informational. The DD was successfully freed using the free command that is listed in the message.

User response:

No action is required.

CKZ00038E **Subtask subtask_number,
module_where_open_issued,
DDNAME(open_failed_ddname),
RC=return_code, RS=reason_code,
open failed**

Explanation:

Db2 Cloning Tool failed to open the specified DD.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00039E **Subtask subtask_number,
module_where_put_issued,
DDNAME(put_failed_ddname),
RC=return_code, RS=reason_code,
put failed**

Explanation:

Db2 Cloning Tool failed to put a record in the specified DD.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00040E Subtask *subtask_number*,
module_where_load_issued,
DDNAME(*last_dd_allocated_for_m*
odule), RC=*return_code*,
RS=*reason_code*, load failed

Explanation:

Db2 Cloning Tool failed to load a utility module.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00041E Subtask *subtask_number*, error
return from RFIDSN call for
masking interface,
RC=*return_code*, RS=*reason_code*

Explanation:

An internal error occurred during work structure preparation for the data masking interface and initialization call.

User response:

If unable to resolve this problem, contact IBM Software Support.

CKZ00042E Subtask *subtask_number*, unable
to initialize the data masking
interface, RC=*return_code*,
RS=*reason_code*

Explanation:

An internal error occurred during work structure preparation for the data masking interface and initialization call.

User response:

If unable to resolve this problem, contact IBM Software Support.

CKZ00043I SUBTASK *subtask_number*,
DSN(*dsname*), SYNCDB2
COMPLETED WITH NO ERRORS

Explanation:

This message is informational.

User response:

No action is required.

CKZ00044I SUBTASK *subtask_number*,
DSN(*dsname*), PAGE
READS=*decimal_number*, PAGE
WRITES *decimal_number*

Explanation:

This message is informational.

User response:

No action is required.

CKZ00045I Subtask *subtask_number*, found
system page entry,
VERS=*change_version*,

SMAP=*space_map_entry*,
PAGE=*page_number*

Explanation:

A system page was found in the header page.

User response:

No action is required.

CKZ00046I Subtask *subtask_number*, system
page found, not first,
PAGE=*page_number*,
DSN=*data_set_name*

Explanation:

A system page was found, but it is not the first page.

User response:

No action is required.

CKZ00047W Subtask *subtask_number*,
DSN(*data_set_name*), SYNCDB2
completed with error(s)

Explanation:

Data set processing completed with errors.

User response:

Review the CKZLOG and CKZPRINT DDs to determine the reason of failure. If unable to determine or resolve this error, contact IBM Software Support.

CKZ00048I Subtask *subtask_number*, replace
page set system page table OBID,
source_OBID, *target_OBID*, page
offset *page_offset*

Explanation:

A table OBID was changed from the source to the target value in a system page.

User response:

No action is required.

CKZ00049E Log apply page call, subtask
subtask_number, RS=*reason_code*,
page=*page_num*

Explanation:

An error occurred during log apply page processing.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00050I Subtask *subtask_number*, RPL
Error Msg: *message_text*

Explanation:

A message is printed from a VSAM macro return code other than zero.

User response:

No action is required.

CKZ00051E Subtask *subtask_number*,
module_name internal error,

**RC=return_code, RS=reason_code,
INFO=additional_information**

Explanation:

An internal error occurred in the module that is listed in the message.

User response:

Contact IBM Software Support.

CKZ00052E *module_name* internal error,
**RC=return_code, RS=reason_code,
INFO=additional_information**

Explanation:

An internal error occurred in the module that is listed in the message.

User response:

Contact IBM Software Support.

CKZ00056W **UNEXPECTED FLAGS=flags, PAGE
BYPASSED**

Explanation:

This message is informational.

User response:

If the target job runs with no errors and the target objects are accessible, ignore this error, otherwise contact IBM Software Support.

CKZ00057I **NON HEADER PAGE, RESET
PGLOGRBA, WAS *old_rba***

Explanation:

This message is informational.

User response:

No action is required.

CKZ00058I **NO MAP TRANSLATE REQUIRED
FOR OBID *current_obid*, ROW
OFFSET *offset_to_start_of_row***

Explanation:

This message is informational.

User response:

No action is required.

CKZ00059I **ZERO PAGE NUMBER, NO
PROCESSING**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00060E **QUEUE READ ERROR,
RS=reason_code**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00061E **QUEUE WRITE ERROR,
RS=reason_code**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00063E **SUBTASK *subtask_number*,
SHOWCB ERROR,
R15=return_code, R0=reason_code**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00064E **SUBTASK *subtask_number*,
access_type, operation,
RC=return_code, RS=reason_code,
ALLOCATION FAILED**

Explanation:

This is a probable system error. Unable to dynamically allocate a file using SVC99.

User response:

Ensure that the file exists and is not allocated to DB2 or another job. Contact IBM Software Support if unable to resolve this error.

CKZ00065E **SUBTASK *subtask_number*,
access_type, operation,
RC=return_code, RS=reason_code,
FREE FAILED,
additional_information**

Explanation:

This is a probable internal error. Unable to dynamically free a file using SVC99.

User response:

Contact IBM Software Support.

CKZ00066E **SUBTASK *subtask_number*,
access_type, operation,
RC=return_code, RS=reason_code,
ALLOCATION FAILED, ERROR
STRING = *error_string***

Explanation:

This is a probable system error. Unable to dynamically allocate a file using SVC99.

User response:

Ensure the file exists and is not allocated to DB2 or another job. Contact IBM Software Support if unable to resolve this error.

CKZ00082E **BUFFER ALLOCATION ERROR,
RC=return_code_from_getmain**

Explanation:

Unable to allocate virtual storage.

User response:

Contact IBM Software Support if unable to resolve this error by increasing the REGION size.

CKZ00083E **SUBTASK** *subtask_number*,
BUFFER ALLOCATION ERROR,
RC=return_code_from_getmain

Explanation:

Unable to allocate virtual storage.

User response:

Contact IBM Software Support if unable to resolve this error by increasing the REGION size.

CKZ00084E **SUBTASK** *subtask_number*,
BUFFER RELEASE ERROR,
RC=return_code_from_getmain

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ00085E **SUBTASK** *subtask_number*, **VSAM**
GET ERROR, R15=return_code,
RPLFDBK=field_from_vsam_RPL,
DSN=data_set_name

Explanation:

This may be an IO error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00086W **SUBTASK** *subtask_number*, **VSAM**
GET Error, R15=return_code,
RPLFDBK=field_from_vsam_RPL,
DSN=data_set_name, DATASET
EXTENSION IS EMPTY

Explanation:

This may be an IO error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00088E **SUBTASK** *subtask_number*, **VSAM**
PUT ERROR, R15=return_code,
RPLFD=field_from_vsam_RPL

Explanation:

This may be an IO error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00089E **Subtask** *subtask_number*, **point**
error, R15=vsam_return_code,
RPLFDBK=vsam_feedback_code

Explanation:

An error occurred during VSAM processing.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ00097E **DATASET** *data_set_name* **HAS AN**
INVALID PARTITION NUMBER

Explanation:

This is a user error. The partition number specified is not within the range supported by DB2.

User response:

Correct the error and resubmit the job.

CKZ00098E **Member of data set specified in DD**
ddname is not available

Explanation:

The member of a PDS or PDSE data set that is specified in the DD *ddname* does not exist.

User response:

Specify the name of an existing member in a PDS or PDSE data set with valid control statements and resubmit the job.

CKZ00100I **FOUND DATASET EXTENSION**
data_set_name

Explanation:

This message is informational. A non partitioned data set extension (other than A001) has been found in a z/OS catalog.

User response:

No action is required.

CKZ00101E **SOURCE DB2 VERSION** *version* **IS**
NOT SUPPORTED

Explanation:

The DB2 version that is listed in the message is not supported.

User response:

Run on a supported DB2 version.

CKZ00103I **Call DB2 source subsystem,**
waiting...

Explanation:

This message is informational. This DB2 call may take several minutes to complete.

User response:

No action is required.

CKZ00107E **SUBTASK** *subtask_number*,
copy_program **COMMAND**
RESPONSE PARSING ERROR
RC=return_code, RS=reason_code

Explanation:

This is an error caused by getting an unexpected value in the message string returned by the copy program.

User response:

Contact IBM Software Support.

CKZ00113I **Messages from RTS processing module will follow**

Explanation:

This message is informational and precedes messages CKZ72000I, CKZ72001W, and CKZ72002E for real-time statistics processing module messages.

User response:

No action is required.

CKZ00114I **End of messages from RTS processing module**

Explanation:

This message is informational. It indicates that all messages from real-time statistics processing module were printed.

User response:

No action is required.

CKZ00130E **Subtask *subtask_number*, space STOP error for one or more spaces and MAX-COPY-RC is 0 or 4, code=*reason_code*, no copies attempted**

Explanation

One or more spaces cannot be correctly stopped. One error message prints for each reason code. The following reason codes can be returned:

- 2: Synchronization error.
- 93: ADVISORY or RESTRICT status was found.
- 218: A status other than STOP, STOPP, RO, or RW was received.

User response

- For code 2, contact IBM Software Support.
- For code 93, ensure that all status values that were added with SET ADVISORY and SET RESTRICT are correct. If they are correct, determine which spaces are in error, and correct them. Run the job with PGM(NONE) until all the status errors have been corrected. No stops or starts are attempted, however all ADVISORY and RESTRICT status values are checked.
- For code 218, determine which spaces are in error and correct.

If copies are desired for those spaces that are not in error, set MAX_COPY_RC=8 in the CKZINI PARMLIB, or

add SET command MAX-COPY-RC(8) and rerun the job.

CKZ00131I **SUBTASK *subtask_number*, DB2 SUBSYSTEM *subsystem*, ALL TARGET | SOURCE SPACES STOPPED**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00132I **SUBTASK *subtask_number* NO SOURCE SPACE STARTS PER AUTO-START-SOURCE-SPACE PARAMETER**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00133I **SUBTASK *subtask_number*, DB2 SUBSYSTEM *subsystem*, ALL TARGET | SOURCE REQUIRED SPACE STARTS COMPLETED**

Explanation:

This message is informational. For the target subsystem, this is all the spaces. For the source subsystem, it is all the spaces that were initially started in RW or RO.

User response:

No action is required.

CKZ00134I **SUBTASK *subtask_number*, DB2 SUBSYSTEM *subsystem*, ATTEMPTING RECOVERY OF SOURCE OBJECTS TO PRE EXECUTE STATUS**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00135I **SUBTASK *subtask_number*, DB2 SUBSYSTEM *subsystem*, *space_type* *database_name.space_name* CHANGE NOT ATTEMPTED, NO RECOVERY REQUIRED**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00137I **SUBTASK *subtask_number*, DB2 SUBSYSTEM *subsystem*,**

space_type
database_name.space_name
*partition*CHANGE NOT
ATTEMPTED, NO RECOVERY
REQUIRED

Explanation:

This message is informational.

User response:

No action is required.

CKZ00138I **SUBTASK *subtask_number*, DB2
SUBSYSTEM *subsystem*, NO
RECOVERY OF TARGET OBJECTS
TO PRE EXECUTE STATUS DUE TO
SIM(N) PARM SPECIFICATION**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00141W **SUBTASK *subtask_number*,
UNABLE TO CONNECT TO SSID
subsystem USING CAF, CANNOT
ISSUE TARGET RECOVERY
COMMANDS**

Explanation:

This message is a warning.

User response:

Contact your system programmer.

CKZ00142I **SUBTASK *subtask_number*, DB2
SUBSYSTEM *subsystem*,
ATTEMPTING RECOVERY OF
TARGET OBJECTS TO PRE
EXECUTE STATUS DUE TO SIM(A)
PARM SPECIFICATION**

Explanation:

This message is informational.

User response:

No action is required.

CKZ00144E **SUBTASK *subtask_number*,
MAX_COPY_RC EXCEEDED WITH
RC=*return_code*, RS=*reason_code***

Explanation:

This error will terminate the source job as requested by MAX_COPY_RC. The Db2 Cloning Tool Table Space Cloning return code is an 8 as MAX_RC = 0. Note that copies already sent to the copy program will continue. No new commands will be issued, however.

User response:

Contact IBM Software Support.

CKZ00153W **SUBTASK *subtask_number*,
MAX_COPY_RC EXCEEDED WITH
RC=*return_code*, RS=*reason_code***

Explanation:

This error will terminate the source job as requested by MAX_COPY_RC. The Db2 Cloning Tool Table Space Cloning return code is a 4 as MAX_RC > 0. Note that copies already sent to the copy program will continue. No new commands will be issued, however.

User response:

Find the DSS error message(s) in CKZPRINT output. If the messages are related to storage or other resource shortages, try reducing the number of DSS commands specified in the DSS_COPY_COMMANDS setting in the PARMLIB. The default is 24. Rerun the source job using a smaller value, such as 4. If the DSS error messages are not related to storage, resolve the DSS error or errors and resubmit the job to copy the failed data set(s).

CKZ00154W **SUBTASK *subtask_number*,
RETURN OF 8 ENCOUNTERED FOR
A DATA SET COPY**

Explanation:

This warning indicates a copy failed and MAX_COPY_RC = 8. Copies continue.

User response:

Resolve the DSS error or errors and resubmit the job to copy the failed data set(s).

CKZ00194I **Index *creator_name*
(*database_name.index_space_name*)
requires a REBUILD, index data
sets will not be copied to target**

Explanation:

An NPI index or all partitions of a partitioned index were excluded from the cloning process because REBUILD is required to be run on the target. If REBUILD-INDEXES-EXECUTE is enabled, indexes will be rebuilt. Otherwise, CKZINTRB DD should be specified in the target job to allow Db2 Cloning Tool to correctly generate statements for the target REBUILD.

User response:

If not already included, specify CKZINTRB DD in the target job JCL. Otherwise, no action is required.

CKZ00196I **Index *creator_name*
(*database_name.index_space_name*)
instance instance(s), requires a
REBUILD, index data sets will not
be copied to target**

Explanation:

Instances of an NPI index or all partitions of a partitioned index were excluded from the cloning process because REBUILD is required to be run on the

target. If REBUILD-INDEXES-EXECUTE is enabled, the index will be rebuilt. Otherwise, CKZINTRB DD should be specified in the target job to allow Db2 Cloning Tool to correctly generate statements for the target REBUILD.

User response:

If not already included, specify CKZINTRB DD in the target job JCL. Otherwise, no action is required.

CKZ00200E *parm_name* must be 1 character

Explanation:

The parameter *parm_name* value is too long or too short.

User response:

Correct the input and resubmit the job. If unable to resolve the error (for example, the error occurred during SYNCDB2 parsing), contact IBM Software Support.

CKZ00201E **KEYWORD** *command_keyword* HAS TOO MANY OPERANDS

Explanation:

This is a user error.

User response:

Specify the correct number for operands for the command.

CKZ00202E **KEYWORD** *keyword* HAS NO OPERANDS

Explanation:

This is a user error.

User response:

Specify the correct number for operands for the command.

CKZ00203E **REQUIRED KEYWORD,** *command_keyword*, MISSING FROM COMMAND

Explanation:

This is a user error.

User response:

Supply the missing keyword.

CKZ00204W **DB2_COMMAND_RESPONSE_WAIT PARM** HAS AN INVALID VALUE, SET COMMAND IGNORED

Explanation:

This is a user error.

User response:

Specify a decimal number from 1 to 999999 as the number of seconds to wait for all DB2 spaces to stop and start. If the PARMLIB value for MAX_RC is set to 4, a default number of seconds will be used and the job will continue.

CKZ00205W **Both IP address and DNS name are specified for** *job_type* TCP/IP server job. IP address will be ignored

Explanation:

When both IP address and DNS name are specified for the source or target TCP/IP server job, DNS name will be used for remote connection and IP address will be ignored.

User response:

Correct the input and resubmit the job.

CKZ00206E *parameter_name* MUST BE SPECIFIED, THERE IS NO DEFAULT

Explanation:

This is a user error.

User response:

Correct the error and resubmit the job.

CKZ00207E **CKZIN OPERAND** *operand_in_CKZIN*, BAD LENGTH

Explanation:

This is a user error.

User response:

Correct the error and resubmit the job.

CKZ00208E **UNRECOGNIZED CKZIN** COMMAND, *command_name*

Explanation:

This is a user error.

User response:

Correct the error and resubmit the job.

CKZ00209E **INVALID FORMAT,** *string_with_parameter_error*

Explanation:

This is a user error.

User response:

Correct the error and resubmit the job.

CKZ00210E *message_text*

Explanation:

This is a user error. The message text provides information about the error.

User response:

Correct the error and resubmit the job.

CKZ00211E **CKZIN COMMAND** *command*, NO OPERAND

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ00212E NO COMMANDS FOUND**Explanation:**

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ00213I END COMMAND FILE SYNTAX CHECKING**Explanation:**

This message is informational.

User response:

No action is required.

CKZ00214W SUBTASK-TERMINATION-WAIT Parm has an Invalid Value; SET Command Ignored**Explanation:**

This is a user error. The default will be used.

User response:

Correct the error and resubmit the job.

CKZ00215W OPTION *option_name* has an invalid value, *value*, defaulting to *default_value***Explanation:**

An invalid value was entered for the *option_name* parameter that is listed in the message.

User response:

Correct the input and resubmit the job or use the default value.

CKZ00216E Non-numeric value found in *numeric_TABLEDEF_field***Explanation:**

The numeric field that is listed in the message has a non-numeric value. This error might be a user error if the SYNCDDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ00217E SIO9 ERROR RETURN, *return_code***Explanation:**

This may or may not be a user error. CKZIN cannot be read.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ00300I Subtask *subtask_number*, Target subsystem *subsystem*, Object not

in Db2 Catalog, *object_type*
object_qualifier1.object_qualifier2

Explanation:

This message is informational.

User response:

No action is required.

CKZ00301I Target cache *object_type* list is empty**Explanation**

The cache list for the object type that is listed in the message is empty for one of the following reasons:

- The cache is not consistent with the current catalog state.
- The TARGET-PREFETCH-DATABASE-LIST sent to the target subsystem to populate the cache contained the names of nonexistent databases.

User response:

Run the job with ENABLE-TARGET-PREFETCH(Y) to populate or refresh the cache on the target system for the required databases.

CKZ00302I *object_type* Cache Populate Complete for database *dbname*, Rows=*rows*, Elapsed Time=*time***Explanation:**

Catalog prefetch has populated the cache. There will be one message for each object type cached. There will be a set of messages for the source, target, or both. *object_type* can be TS, TP, TB, or IX; *dbname* is the database name; *rows* is the decimal number of rows that was hashed; *time* is time spent caching catalog table.

User response:

No action is required.

CKZ00303I Database *database_name* was cached**Explanation:**

This message is informational.

User response:

No action is required.

CKZ00304E SUBTASK *subtask_number*, NO ENTRY IN SYSIBM.SYSAUXRELS FOR AUX TABLE *subsystem* *aux_table_creator.aux_table_name***Explanation:**

This is a DB2 catalog inconsistency.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00305E Subtask *subtask_number*, No Entry in SYSIBM.SYSXMLRELS for XML Table *subsystem* *aux_table_creator.aux_table_name*

Explanation:

This error indicates a Db2 catalog inconsistency.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00306E Subtask *subtask_number*, Subsystem *subsystem*, Relname = *relation_name*, Catalog Inconsistency, Cannot Find Primary Index

Explanation:

This error indicates a Db2 catalog inconsistency.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00307E DB *database_name* was not found in source catalog

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00308E No *object_type* were found in catalog for *space_type* *database_name.space_name* extracted by LISTDEF utility for cloning

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00309E *space_type* *database_name.space_name* extracted by LISTDEF utility for cloning was not found in source catalog

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00310E No *object_type* were found in catalog for *parent_object_type* *object_qualifier_1.object_qualifier_2*

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00311I Subtask *subtask_name*, DDL Create, object exists, processing continues, *DDL_statement*

Explanation:

This object already exists on the target.

User response:

No action is required.

CKZ00312W TABLE *table_creator.table_name* AND COLUMN *column_name*, DEFAULT COLUMN = *col_default_from_SYSCOLUMNS*, BUT NO ENTRY FOUND IN SYSIBM.SYSSEQUENCESDEP

Explanation:

This is a DB2 catalog inconsistency. The column indicated it was an identity column, but it was not found in the identity column catalog table.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00313W TABLE *table_creator.table_name* AND COLUMN *column_name*, DEFAULT COLUMN = *col_default_from_SYSCOLUMNS*, BUT NO ENTRY FOUND IN SYSIBM.SYSSEQUENCES FOR ID = *BSEQUENCEID_from_SYSEQUENCESDEP*

Explanation:

This is a DB2 catalog inconsistency. The column indicated it was an identity column, but it was not found in the identity column catalog table.

User response:

Resolve the inconsistency and resubmit the job.

CKZ00314I Source | Target Db2 current APPLCOMPAT level: *current_applcompat_level*, catalog level *catalog_level*

Explanation:

This informational message displays current APPLCOMPAT and catalog levels for the given Db2 subsystem. *current_applcompat* is the current APPLCOMPAT level, such as V12R1M500). *catalog_level* is the catalog level, such as V12R1M500.

User response:

No action is required.

CKZ00315I Subtask *subtask_number*, DDL DROP, object does not exist,

processing continues,
DDL_statement

Explanation:

This object does not exist on the target.

User response:

No action is required.

CKZ00400I BEGIN PAGE PROC, OBJ: *object*
SEQ#: SEQNR FLG:*flag_byte* DSN:
data_set_name

Explanation:

This message is informational.

User response:

No action is required.

CKZ00401I DICTIONARY PAGE DB.TS
REPLACEMENT,
target_database.target_table_spac
e,
source_database.source_table_spa
ce

Explanation:

This message is informational.

User response:

No action is required.

CKZ00402I HEADER PAGE CHANGES --- TBL
OBID *old_table_object_id*
new_table_object_id, RECORD
DESC
old_record_descriptor_object_id
new_record_descriptor_object_id,
IX OBID *old_index_object_id*
new_index_object_id

Explanation:

This message is informational.

User response:

No action is required.

CKZ00403I RESET PGLOGRBA, WAS
old_table_LOGRBA

Explanation:

This message is informational.

User response:

No action is required.

CKZ00404I HEADER PAGE CHANGES --- DBID/
PSID *old_dbid new_dbid, logrba*
RESET, SSID *old_ssid new_ssid*,
OBID *old_obid new_obid*

Explanation:

This message is informational.

User response:

No action is required.

CKZ00405I SG/VCAT CHANGES --- SG:
old_storage_group
new_storage_group, VCAT:
old_VCAT new_VCAT

Explanation:

This message is informational.

User response:

No action is required.

CKZ00406I INDEX DATA PAGE, OBID CHANGE
old_obid new_obid, PAGE OFFSET
offset_value

Explanation:

This message is informational.

User response:

No action is required.

CKZ00407I BEGIN PROCESSING INDEX
DIRECTORY PAGE

Explanation:

This message is informational.

User response:

No action is required.

CKZ00408I UNFORMATTED INDEX PAGE,
DISCARDED

Explanation:

This message is informational.

User response:

No action is required.

CKZ00409I BEGIN PROCESSING INDEX
SPACE MAP

Explanation:

This message is informational.

User response:

No action is required.

CKZ00410I BEGIN PROCESSING INDEX DATA
PAGE, IPFLAGS-
flags_in_the_page_header

Explanation:

This message is informational.

User response:

No action is required.

CKZ00411I LOB DATA PAGE

Explanation:

This message is informational.

User response:

No action is required.

CKZ00412I LOB HIGH LEVEL SPACE MAP
PAGE

Explanation:

This message is informational.

User response:

No action is required.

CKZ00413I LOB LOW LEVEL SPACE MAP PAGE

Explanation:

This message is informational.

User response:

No action is required.

CKZ00414I LOB MAP PAGE

Explanation:

This message is informational.

User response:

No action is required.

CKZ00415I UNIDENTIFIED LOB PAGE

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00416I REPLACE PAGE SET DATA PAGE
MAP ID, *old_map_id new_map_id*,
PAGE OFFSET *page_offset***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00417I HEADER PAGE,
PGCOMB=*flags_in_pageset_header*,
r,
FLAGS=*flags_in_pageset_header*,
HPGFLAGS=*flags_in_pageset_header***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00418I HEADER PAGE, LARGE,
PGCOMB=*flags_in_pageset_header*,
r,
FLAGS=*flags_in_pageset_header*,
HPGFLAGS=*flags_in_pageset_header***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00419I BEGIN PROCESSING DATA PAGE,
#MAPIDS: *number_of_map_ids***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00420I BEGIN PROCESSING SEGMENTED
SPACE MAP WITH SEGSIZE
*segment_size***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00421I SPACE MAP CHANGES --- OBID
old_map_id new_map_id, PAGE
OFFSET *page_offset***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00422I SINGLE TABLE OBID CHANGE,
*old_obid new_obid***

Explanation:

This message is informational.

User response:

No action is required.

CKZ00423I SYSTEM PAGE FOR ODBRECS

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00424I DICTIONARY PAGE, NOT HEADER,
IGNORED**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00425I NON SEGMENTED SPACE MAP
PAGE, NO OBIDS TO TRANSLATE**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ00426I HASH SI UTS
segment_info_UTS_entry_num
Table OBID Change -
source_table_OBID_in_hex,
*target_table_OBID_in_hex***

Explanation:

This message is informational.

User response:

No action is required.

CKZ00427I **HASH Bucket**
hash_bucket_entry_num Table
OBID Change -
source_table_OBID_in_hex,
target_table_OBID_in_hex

Explanation:

This message is informational.

User response:

No action is required.

CKZ00428I **Begin page proc, obj: obj_type**
seq#: block_number **flg: page_flag**
DSN: data_set_name **PGEND:**
page_tail

Explanation:

Page processing started.

User response:

No action is required.

CKZ00429I **RBRBA=recover_base_rba,**
LEVEL=level_id,
PLEVL=previous_level_id,
CLRSN=image_copy_bit_update_LR
SN,
RBLP=recovery_base_log_point,
CKPT=last_checkpoint_time_used

Explanation:

This is an informational tracing message.

User response:

No action is required.

CKZ00430I **Possible CIHDR**

Explanation:

This is an informational tracing message. The CI header was met.

User response:

No action is required.

CKZ00431I **CIHDR, Reset LOGRBA,**
logrba_value, Page Offset *offset*

Explanation:

This is an informational tracing message. The CI header and LOGRBA are changed.

User response:

No action is required.

CKZ00432I **CIHDR, Change IX OBID, src_obid,**
trg_obid

Explanation:

This is an informational tracing message. The CI header and index OBID are changed.

User response:

No action is required.

CKZ01000E **????????? SUPPORT MODULE**
MISSING | ?????????? DD
STATEMENT MISSING

Explanation:

During program start, either required modules or DD statements were not discovered.

User response:

For missing modules, ensure the proper //STEPLIBs are available. For missing DDs, add the appropriate DD statement to the execution JCL.

CKZ01002E **UNRECOGNIZABLE COMMAND (N)**
ccccccccc

Explanation:

During command interpretation, command ccccccccc could not be identified. As this situation can be determined in several places, sequence "n" isolates the point the error was detected.

User response:

Check the spelling of the command.

CKZ01003E **COMMAND HAS NO OPERANDS**

Explanation:

During command interpretation, the command preceding this message did not have any operands.

User response:

Most commands have operands. Ensure continuation indicators are present if the command was continued onto a second line.

CKZ01005E **FATAL ERRORS HAVE OCCURRED**
DURING //CKZINI PROCESSING.

Explanation:

During program start, one or more problems occurred during decoding of the //CKZINI member. As the //CKZINI provides vital information for Db2 Cloning Tool, the program cannot continue execution.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ01006E **DB2 CLONING TOOL MUST**
EXECUTE AS A Z/OS APF
AUTHORIZED PROGRAM

Explanation:

During program start, Db2 Cloning Tool determined it was not z/OS APF authorized. Due to restraints set forth by z/OS, numerous functions will not work, thus the program cannot continue execution.

User response:

Ensure that the library from which Db2 Cloning Tool is executing is z/OS APF authorized.

CKZ01007E INI AND MODULE RELEASES DO NOT MATCH.

Explanation:

During program start, Db2 Cloning Tool determined that the release in the CKZINI member does not match the internal release. Processing terminates.

User response:

Ensure that INIMERGE has been run.

CKZ01009I DB2 CLONING TOOL EXECUTION COMPLETE. HIGHEST RETURN CODE WAS n.

Explanation:

Program termination message. "n" is the highest return encountered during processing of the commands. See other messages to determine the cause of a non-zero return code.

User response:

No action is required.

CKZ01010E UNABLE TO DETERMINE MASTER CATALOG

Explanation:

During program start, a problem occurred attempting to determine the name of the master catalog. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ01011E SWAREQ FAILED; R15=nnnnnnnn BLOCK REQUESTED=block

Explanation:

An SWAREQ has failed for the indicated block. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ01020I PROGRAM: name info

Explanation:

Displays the maintenance level of a loaded Db2 Cloning Tool program.

User response:

No action is required.

CKZ01099E ABEND DURING DB2 CLONING TOOL PROCESSING

Explanation:

During execution of a command within Db2 Cloning Tool, an abend took place that the command did not rectify.

User response:

Ensure that all parameters on the named command are coded correctly, that external data sets used in the command are valid and try again. If the problem persists, contact IBM Software Support. Ensure all software run information is available: this includes the original JCL and control statements used to invoke Db2 Cloning Tool, and all the spooled output from its execution.

CKZ02001I hh:mm:ss COPY PROCESS COMPLETED; RETURN CODE=nnn

Explanation:

COPY command processing message. For the 'completed' message, if the return code is non-zero, check other messages for errors and/or warnings.

User response:

No action is required.

CKZ02003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the specified data set.

User response:

No action is required.

CKZ02004E DDNAME MISSING: ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ02005E ALLOCATION FAILED FOR DSN: datasetname

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ02005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ02006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ02007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ02008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ02009E ERROR ACCESSING JOURNAL
FILE; LOC=*location* | ERROR
ACCESSING BASE JOURNAL FILE;
LOC=*location***

Explanation:

A VSAM error occurred accessing the journal file or the base journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02010E DUPLICATE JOURNAL ENTRY;
LOC=*location* | DUPLICATE BASE
JOURNAL ENTRY; LOC=*location***

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02011E JOURNAL CONTROL RECORD NOT
FOUND | BASE JOURNAL
CONTROL RECORD NOT FOUND |
JOURNAL *record_type* RECORD
NOT FOUND | BASE JOURNAL
record_type RECORD NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02012E BASE JOURNAL CONTROL
RECORD IS WRONG VERSION |
BASE JOURNAL *record_type*
RECORD IS WRONG VERSION |
JOURNAL VOLP RECORD IS
WRONG VERSION**

Explanation:

The wrong version of a record is in the journal file or in the base journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02013E RECORD COUNT IS ZERO; LOC=
location | *record_type* COUNT
MISMATCH; RECORDS READ=
number_read CONTROL RECORD
COUNT= *count* LOC= *location***

Explanation:

The count of records in the base journal file is incorrect. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02014E THE BASE COPY PROCESS DID
NOT COMPLETE SUCCESSFULLY**

Explanation:

The base COPY command that runs before the current COPY command did not complete successfully. Processing terminates.

User response:

Verify and correct the parameters of the previous base COPY command and rerun the command.

**CKZ02016E SIMULATE MUST BE SPECIFIED
BECAUSE THE BASE COPY WAS
SIMULATE**

Explanation:

The base COPY command was run with the SIMULATE option, but the current COPY command was run without the SIMULATE option. Processing terminates.

User response:

Rerun this COPY command with the SIMULATE option, or rerun the base COPY command without the SIMULATE option.

**CKZ02017E THE DDNAME IS EMPTY OR HAS
BEEN DUMMIED, DDNAME: ddn**

Explanation:

No records were read from the ddname specified for a keyword. Processing terminates.

User response:

Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

**CKZ02018E THE DDNAME INPUT HAS
EXCEEDED THE CURRENT
CAPACITY, DDNAME: ddn**

Explanation:

The number of entries read from the ddname exceeded the current capacity. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02019E THE keyword DOES NOT HAVE A
LRECL OF 80, DDNAME: ddn**

Explanation:

The data set allocated to the ddname for the keyword does not have a LRECL of 80. The LRECL of this data set must be 80.

User response:

Change the data set allocated to the ddname to have a LRECL of 80.

**CKZ02020I TARGET VOLUMES WILL BE
CLIPPED WHEN THE VOLOPTIIONS
TARGETOFFLINECLIP COMMAND
IS EXECUTED**

Explanation:

The target volumes will not be clipped by the COPY command due to the use of the VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN keywords. The target volumes will be clipped when the VOLOPTIIONS TARGETOFFLINECLIP command is executed.

User response:

No action is required.

**CKZ02021I SOURCE USER CATALOGS WILL BE
BACKED UP WHEN THE
UCATOPTIONS BACKUP
COMMAND IS EXECUTED**

Explanation:

The source user catalogs will not be backed by the COPY command due to the use of the USERCATALOGS-NOBACKUP keyword. The source user catalogs will be backed up when the UCATOPTIONS BACKUP command is executed.

User response:

No action is required.

**CKZ02022I VOLSER: volser LOGICAL NUMBER
OF CYLINDERS: nnnnnnnn IS LESS
THAN PHYSICAL NUMBER OF
CYLINDERS: nnnnnnnn**

Explanation:

The identified volume has a logical size, from the Format 4 DSCB, that is less than the physical size, from the DCE. The logical size will be used for pairing this volume. This may lead to a condition where there are not enough target volumes available of the correct size to pair with all the source volumes.

User response:

No action is required. ICKDSF can be used to make the logical size equal to the physical size.

**CKZ02024E FCTOXRCPRIARY CANNOT BE
USED. REASON: FASTREP IS NOT
SUPPORTED | DFSMSDSS HAS NO
SOFTWARE SUPPORT FOR
FCTOXRCPRIARY**

Explanation:

The DATA-MOVER options FASTREP or FCTOXRCPRIARY cannot be used for the COPY command on this system. Processing terminates.

User response:

Verify that the FASTREP and FCTOXRCPRIARY options of the DFSMSDSS COPY command are supported on the system. If not supported, remove these options. If some of the target volumes are XRC primary volumes, then use FASTREP(NONE) in the DATA-MOVER option of the COPY command.

**CKZ02028E THE BASE JOURNAL APPEARS TO
HAVE BEEN UPDATED BY A PRIOR
RUN OF THE COPY COMMAND;
UNABLE TO CONTINUE
PROCESSING**

Explanation:

The base journal was modified by a prior run of the COPY command. Processing terminates.

User response:

Remove the base journal and rerun the base COPY command.

CKZ02029E UNEXPECTED *record_type* RECORD IN BASE JOURNAL LOC=*location*

Explanation:

An unexpected record was found in the base journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02029I Log apply process was unable to connect to the server

Explanation:

The log-apply process failed to connect to the server to output for output.

User response:

Ensure that the Db2 Cloning Tool server is active.

CKZ02030I DSS LEVEL=X'nnnnnnnn'

Explanation:

The level of DSS returned by ADRMCLVL.

User response:

No action is required.

CKZ02031I NUMBER OF *record_type* RECORDS ADDED TO BASE JOURNAL: *number* | NUMBER OF *record_type* RECORDS UPDATED IN BASE JOURNAL: *number*

Explanation:

This informational message displays the number of *record_type* records that were added or updated to or in the base journal file.

User response:

No action is required.

CKZ02032I SETTING BASE JOURNAL TO TARGET-UCATS-ON-TARGET-VOLUMES(Y) BECAUSE THIS RUN HAS TARGET-UCATS-ON-TARGET-VOLUMES(Y)

Explanation:

The TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword value was added to the base journal file because the current base COPY command has the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option set.

User response:

No action is required.

CKZ02033I SETTING TARGET-UCATS-ON-TARGET-VOLUMES(Y) BECAUSE THE BASE JOURNAL HAS TARGET-UCATS-ON-TARGET-VOLUMES(Y)

Explanation:

The TARGET-UCATS-ON-TARGET-VOLUMES(Y) option was set for this COPY command because the base journal file contains the TARGET-UCATS-ON-TARGET-VOLUMES (Y) value from the base COPY command.

User response:

No action is required.

CKZ02034I TARGET CATALOGS ON TARGET VOLUMES

Explanation:

There are target catalogs on the target volumes.

User response:

No action is required.

CKZ02035I SOURCE | TARGET CATALOG: *catalog_name* HAS VOLSER: *volser* IN THE USERCATALOGS KEYWORD. THE MASTER CATALOG HAS A DIFFERENT VOLSER: *volser*. THE USERCATALOGS VOLSER WILL BE USED

Explanation:

The source or target user catalog's VOLSER will be used because the source or target catalog on the source or target volume is defined in a USERCATALOGS keyword, and the master catalog has different VOLSER.

User response:

No action is required.

CKZ02036E THE VOLSER SPECIFIED FOR A SOURCE USERCATALOG IS NOT A SOURCE VOLSER | THE VOLSER SPECIFIED FOR A TARGET USERCATALOG IS NOT A TARGET VOLSER

Explanation

This message is issued under one of the following circumstances:

- When the TARGET-UCATS-ON-TARGET-VOLUMES(Y) and PGM(NONE) are used for a COPY command, and the VOLSER for a source user catalog is not a source VOLSER.
- When the TARGET-UCATS-ON-TARGET-VOLUMES(Y) is used for a COPY command and the VOLSER for a target user catalog is not a target VOLSER.

Processing terminates.

User response:

Correct the USERCATALOGS option and options for the source or target VOLSERs

CKZ02037E A VOLSER HAS BEEN SPECIFIED FOR A TARGET CATALOG IN A

USERCATALOGS ENTRY. THIS IS NOT ALLOWED WHEN THE TARGET-UCATS-ON-TARGET-VOLUMES KEYWORD IS NOT BEING USED | A VOLSER HAS BEEN SPECIFIED FOR A TARGET CATALOG IN A USERCATALOGS ENTRY. THIS IS NOT ALLOWED WHEN THE DATA-MOVER PGM VALUE IS NOT NONE

Explanation:

A VOLSER was specified for a target catalog in a USERCATALOGS entry. This is not allowed when the TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword is not being used, or when the DATA-MOVER PGM value is not NONE. Processing terminates.

User response:

Either add the TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword to the COPY command, or set the DATA-MOVER(PGM(NONE)) option.

CKZ02038E **CATALOG: *catalog_name* ON TARGET VOLUME: *volser* CATALOG IS NOT SPECIFIED AS A TARGET IN THE BASE JOURNAL USERCATALOGS KEYWORD | CATALOG: *catalog_name* ON TARGET VOLUME: *volser* CATALOG IS NOT SPECIFIED AS A TARGET IN THE USERCATALOGS KEYWORD**

Explanation:

A catalog is on target volume, but this catalog is not specified as either a target in the base journal file's USERCATALOGS keyword or as a target in the USERCATALOGS keyword. Processing terminates.

User response:

If the problem is with the base journal, verify the base journal file from base COPY command and correct the input if necessary. If the problem is with the USERCATALOGS keyword, define the user catalog in the USERCATALOGS keyword. If unable to resolve this problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ02039E **INTERNAL ERROR; LOC=*location***

Explanation:

This is internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02040E **INVALID VALUE FOUND FOR item IN keyword RECORD: *value***

Explanation:

An invalid value has been found for an item in a record in the data set allocated to the ddname for the keyword. The record is printed. Processing terminates.

User response:

Correct the value for the item in the record to have a valid value.

CKZ02041E ***program_name* UNEXPECTED RESULTS; *error_text***

Explanation:

An unexpected condition occurred calling the program that is listed in the message. *error_text* has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02042I **NO TARGET CATALOG ON TARGET VOLUMES**

Explanation:

The TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword was defined for the COPY command, but there are no ICF catalogs on the target volumes.

User response:

No action is required.

CKZ02043I **ANTRQST LEVEL=*nn*; ESSRVCS LEVEL=*nnn***

Explanation:

The level returned by ANTRQST REQUEST=LEVEL.

User response:

No action is required.

CKZ02044W **ANTRQST LEVEL NOT SUPPORTED; LEVEL=*nn***

Explanation:

For FlashCopy support, the level must be greater than four (4). For SnapShot support, the level must be greater than one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response:

Check with your system programmer for upgrading the system.

CKZ02045E **ANTMAIN NOT ACTIVE**

Explanation:

An ANTRQST request failed because the system task, ANTMAIN, is not active. Processing terminates.

User response:

Start the system task, ANTMAIN.

CKZ02046E **ANTRQST LEVEL DOES NOT SUPPORT *function***

Explanation:

The COPY command has keywords or parameters specified that request use of the identified function but the ANTRQST level does not support that function. Processing terminates.

User response:

Check with your system programmer for upgrading the system.

**CKZ02047E ANTRQST ERROR; request type
RETURN CODE=nnnn X'hhhh'
REASON CODE=nnnn X'hhhh'
LOC=lllll VOLSER=volser**

Explanation:

An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02048I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation:

Informational message indicating how COPY will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

**CKZ02049W keyword SPECIFIED BUT WILL
NOT BE USED DUE TO reason**

Explanation:

The indicated keyword was specified but will not be used due to the indicated reason. Processing continues.

User response:

Correct the keyword specifications.

**CKZ02050E ERROR IN PARAMETERS FOR
keyword**

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ02051E REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ02052E REQUIRED INI SECTION/TOKEN
MISSING: SECTION=section
TOKEN=token | REQUIRED INI
VALUE MISSING FOR
SECTION=section TOKEN=token |
INVALID INI VALUE FOR
SECTION=section TOKEN=token**

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

**CKZ02053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED | TOKEN:
token MAXIMUM LENGTH: nnn
EXCEEDED**

Explanation:

The operand entered for a keyword or CKZINI token exceeded the maximum length allowed. nnn is the maximum allowed length. Processing terminates.

User response:

Correct the length of the keyword's operand or the token's value.

**CKZ02054E KEYWORD HAS MORE THAN n
OPERANDS; ONLY n ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only the indicated number of operands is permitted. Processing terminates.

User response:

Correct the keyword to use the correct number of operands.

**CKZ02055E STORAGE GROUPS NOT
SUPPORTED WITH DATA-MOVER
PROGRAM NONE**

Explanation:

The DATA-MOVER program was specified as 'NONE'. Storage group names/masks were specified for the source and/or target volume serials. Processing terminates.

User response:

Correct the DATA-MOVER program specified, or, use keywords FROM-VOLSER/TO-VOLSER for the volume serials.

**CKZ02056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ02057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

**CKZ02058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ02059E KEYWORD: keyword1 CANNOT BE
USED WITH keyword2**

Explanation:

The first indicated keyword cannot be used with the second indicated keyword. Processing terminates.

User response:

Either remove the first indicated keyword or change the second indicated keyword to be compatible with the first indicated keyword.

**CKZ02060E UCBSCAN ERROR; RETURN
CODE=nn REASON CODE=nn |
UCBINFO ERROR; RETURN
CODE=nn REASON CODE=nn**

Explanation:

An error occurred using UCBSCAN or UCBINFO. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ02061E CKZ01SMF ERROR; RETURN
CODE=nnnn LOC: llllll entry**

Explanation:

An error occurred using CKZ01SMF to obtain SSI information for the 'entry'. llllll is the internal location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ02062E *multiple possible messages; see
Explanation***

Explanation**NO STORAGE GROUPS RETURNED BY SSI**

The COPY command was requested to determine volume serial numbers by storage group names. SSI did not return any storage groups to CKZ01SMF. Processing terminates.

NO VOLUME SERIALS RETURNED BY SSI

The COPY command was requested to check the SMS status of volume serials. SSI did not return any volume serials to CKZ01SMF. Processing terminates.

User response:

Check that storage groups are defined on the system. If unable to determine the cause of this message, contact IBM Software Support. Have available the listing containing this message.

**CKZ02063E EXPLICIT SOURCE STORAGE
GROUP NOT FOUND | EXPLICIT
VOLSER NOT FOUND ONLINE |
MASK FOR SOURCE NOT
RESOLVED**

Explanation:

The indicated 'entry' for the keyword was not matched. For VOL, the indicated volser, or, the volser derived from a storage group, was not found. For STG, the indicated storage group was not found.

User response:

For an error with a source keyword, correct the keyword specification, or, ensure that all source volumes are online.

**CKZ02063W NO VOL/STG MATCH FOUND;
KEYWORD: keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was not matched. For VOL, the indicated volser, or, the volser derived from a storage group, was not found. For STG, the indicated storage group was not found. Processing continues. If the keyword involved source volumes, one of the CKZ02063E messages will be issued.

User response:

No action is required.

**CKZ02064E INVALID VOLSER: volser IN
KEYWORD: keyword**

Explanation:

The volume serial number specified is invalid. Processing terminates.

User response:

Correct the volser specification.

CKZ02065E STORAGE GROUP REFERENCED AS BOTH SOURCE AND TARGET: storagegroup name

Explanation:

The storage group was used in both the FROM-STORAGEGROUP and TO-STORAGEGROUP keywords. Processing terminates.

User response:

Correct the storage group specification.

CKZ02066E VOLSER REFERENCED AS BOTH SOURCE AND TARGET: volsr

Explanation:

The volume serial was used in both the FROM-VOLSER and TO-VOLSER keywords. Processing terminates.

User response:

Correct the volume serial specification.

CKZ02067W NO VOLUME SERIALS FOR STORAGE GROUP: storage group

Explanation:

SSI did not return any volsers for the storage group to CKZ01SMF. Processing continues.

User response:

No action is required.

CKZ02068E UNMATCHED ENTRIES IN KEYWORD: keyword

Explanation:

For USERCATALOGS, there must be a source BCS followed by a target BCS. An uneven number of BCS's was specified. For VOLPAIRS, there must be a source volume serial, target volume serial. An uneven number of entries was specified. For VOLPAIRSDEVN, there must be a source volume serial, target volume serial, target device number. Unmatched entries were found. Processing terminates.

User response:

Correct the keyword specification.

CKZ02069E UNRESOLVED SYMBOL IN KEYWORD: keyword R15=nnnn ERR=error text

Explanation:

A symbol was used in the keyword. CKZ01KSS was unable to resolve the symbol. Processing terminates.

User response:

Define the symbol or remove the symbol.

CKZ02070E SECURITY PRODUCT DENIED ACCESS TO DSN: datasetname

Explanation:

The dsn indicated is not authorized for alter by your security product. If the RACF profile that is associated is returned, it will be displayed. Processing terminates.

User response:

Change the dsn to one you can use, or, have your security administrator give you 'ALTER' authority to the data set.

CKZ02071E RACROUTE ERROR; SAF RC=nnnn RACF RC=nnnn RACF REASON CODE=nnnn

Explanation:

An unexpected return code from SAF or RACF occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ02072E MORE SOURCE VOLSERS THAN TARGET VOLSERS

Explanation:

More volsers were found in the FROM-VOLSER or FROM-STORAGEGROUP keywords than were found in the TO-VOLSER or TO-STORAGEGROUP keywords. Processing terminates.

User response:

Correct the FROM- or TO- specifications. There must be at least as many target volumes as source volumes.

CKZ02073E DATA-MOVER PGM NOT RECOGNIZED; PROGRAM=program

Explanation:

A program was specified in the DATA-MOVER(PGM(..)) keyword. The program is not recognized. Processing terminates.

User response:

Correct the program specification.

CKZ02074E USERCATALOG USED AS BOTH SOURCE AND TARGET: BCS dsname

Explanation:

The BCS dsname was designated as a source in one pair and as a target in another pair. Processing terminates.

User response:

Correct the USERCATALOGS specification.

CKZ02075E NO VOLUME SERIALS SELECTED FOR source/target

Explanation:

No volume serials have been found for Db2 Cloning Tool to use as source volsr, or, no volume serials have

been found for Db2 Cloning Tool to use as target volser. Processing terminates.

User response:

For 'SOURCE', check if volsers specified in the FROM-keyword have been removed by use of the EXCLUDE-FROM-keyword. For 'TARGET', check if volsers specified in the TO-keyword have been removed by use of the EXCLUDE-TO-keyword.

CKZ02076E DEVICE TYPE AND MODEL NOT FOUND FOR volser

Explanation:

IOSCDs for the volser did not return information needed to pair source with target volsers.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02077E EXCLUDE NOT ACCEPTED WITH DATA-MOVER PROGRAM NONE

Explanation:

EXCLUDE-FROM-VOLSER and EXCLUDE-TO-VOLSER are not accepted with DATA-MOVER program 'NONE'. Processing terminates.

User response:

Remove the EXCLUDE keyword.

CKZ02078E VOLPAIRSDEVN ONLY ACCEPTED WITH DATA-MOVER PROGRAM NONE or PPRC.

Explanation:

Keyword VOLPAIRS was used, but, the DATA-MOVER program is not 'NONE'. Keyword VOLPAIRSDEVN was used, but, the DATA-MOVER program was neither 'NONE' nor 'PPRC'. Processing terminates.

User response:

Correct the keyword specifications.

CKZ02079E TARGET DEVICE NUMBER nnnn IS CURRENTLY ONLINE; VOLSER: volser

Explanation:

Keyword VOLPAIRSDEVN was used, but, a specified target device number is online. Processing terminates.

User response:

Check the VOLPAIRSDEVN specification. The target volume serials should be offline to the executing system and have the paired source volume serial in the label.

CKZ02080E TARGET DEVICE NUMBER nnnn PREVIOUSLY SELECTED FOR VOLSER: previous volser CURRENT VOLSER: current volser

Explanation:

The same device number was used for more than one target volume serial. Processing terminates.

User response:

Correct the VOLPAIRSDEVN specification.

CKZ02081E VOLUME volser DEVICE NUMBER nnnn HAS NO chpid PATHS AVAILABLE

Explanation:

The indicated device has no paths available for Db2 Cloning Tool to use to access the volume. Processing terminates.

User response:

Ensure that at least one chpid is online for each target device specified.

CKZ02082E TARGET DEVICE NUMBER: nnnn IS NOT AN ECKD DEVICE

Explanation:

The indicated device does not appear to support ECKD commands. Processing terminates.

User response:

Ensure that only supported devices are specified.

CKZ02083E FASTREP PARAMETER NOT RECOGNIZED; PARM=parameter

Explanation:

The parameter specified in the DATA-MOVER(FASTREP(...)) keyword is not recognized. Processing terminates.

User response:

Correct the parameter specification.

CKZ02084E OFFLINE SOURCES NOT ACCEPTED WITH DATA-MOVER PROGRAM pgm

Explanation:

SOURCEOFFLINE(Y) was specified with DATA-MOVER(PGM(ADRDSSU)) specified (or defaulted) or DATA-MOVER(PGM(EMCSNAP)) specified. The source volumes must be online for ADRDSSU and EMCSNAP. Processing terminates.

User response:

Either remove the SOURCEONLINE keyword, or specify DATA-MOVER(PGM(NONE)).

CKZ02085I DSNS FOR KEYWORD: keyword list of dsns

Explanation:

Parsing found the listed dsns for the keyword.

User response:

No action is required.

CKZ02086I STORAGE GROUPS/MASKS FOR KEYWORD: keyword

Explanation:

Parsing found the listed storage groups/masks for the keyword.

User response:

No action is required.

CKZ02087I **nnnnn VOLUMES OR MASKS FOR KEYWORD: keyword nnnnn VOLUMES DERIVED FOR KEYWORD: keyword nnnnn VOLUMES RESOLVED FOR KEYWORD: keyword nnnnn VOLUME PAIRS FOR KEYWORD: keyword nnnnnn VOLUMES/ DEVICES FOR KEYWORD: keyword list of volsers**

Explanation:

Parsing found the listed volsers/masks for a keyword, or, derived the listed volsers from a storage group keyword. nnnnn is the number of volume serials. The third format, RESOLVED, indicates the number of volume serials found online for the associated keyword.

User response:

No action is required.

CKZ02088I **VOLUME SERIALS TO BE USED FOR SOURCE | VOLUME SERIALS TO BE USED FOR TARGET list of volsers**

Explanation:

The listed volsers will be used during Db2 Cloning Tool COPY processing.

User response:

No action is required.

CKZ02089I **TARGET VOLUMES WILL NOT BE CHECKED FOR EMPTY**

Explanation:

Informational message.

User response:

No action is required.

CKZ02090I **TARGET VOLUMES WILL BE CHECKED FOR EMPTY**

Explanation:

Informational message.

User response:

No action is required.

CKZ02091I **VALIDATING KEYWORD: keyword**

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

CKZ02092E **VOLUME volses FROM BASE JOURNAL IS NOT DEFINED IN THE SYSTEM | VOLUME volses FROM BASE JOURNAL IS NOT DEFINED AS SOURCE VOLUME**

Explanation:

The BASE-JOURNAL-DDN parameter is used, but the volume volses used as a target volume in the previous COPY command is not specified as a source volume in the current COPY command, or was not found in the system. Processing terminates.

User response:

If the volume was not found in the system, verify that the volume is online. If the volume is not defined as a source volume, then verify the input parameters for the COPY command. If unable to resolve this error, contact IBM Software Support.

CKZ02093I **THE FOLLOWING VOLUMES ARE NOT FROM BASE JOURNAL AND WERE EXCLUDED: list_of_volsers**

Explanation:

The BASE-JOURNAL-DDN parameter is used, but the listed source volumes were not used as targets for a previous COPY command. Therefore, the volumes were excluded.

User response:

No action is required.

CKZ02101I **hh:mm:ss VOLUME COPY STARTED - PROGRAM REV=rrr (** SIMULATION **) | hh:mm:ss VOLUME COPY COMPLETED; RETURN CODE=nnn**

Explanation:

VOLUME PAIRING processing message.

User response:

No action is required.

CKZ02104E **OPEN FAILED FOR DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ02107W **ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:
Please report this message to IBM Software Support.

**CKZ02108E UNABLE TO LOAD PROGRAM:
 program name**

Explanation:
The indicated program name was not found.
Processing terminates.

User response:
Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ02109E ERROR ACCESSING JOURNAL
 FILE; LOC=lllll**

Explanation:
A VSAM error occurred accessing the journal file.
Processing terminates.

User response:
See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02110E DUPLICATE JOURNAL ENTRY;
 LOC=lllll**

Explanation:
A duplicate record was detected. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02111E JOURNAL VOLUME PAIR
 RECORD(S) NOT FOUND |
 JOURNAL UCAT PAIR RECORD(S)
 NOT FOUND**

Explanation:
An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02112E JOURNAL UCAT PAIR RECORD IS
 WRONG VERSION**

Explanation:
The journal record does not match the expected format. The record is printed. Processing terminates.

User response:
Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ02130I ADRDSSU TASK COMPLETED;
 RETURN CODE=nn SYSOUT
 DD=ddname**

Explanation:
An ADRDSSU task has ended.

User response:
No action is required if the RETURN CODE is zero. If the RETURN CODE is not zero, check the indicated sysout file for warning or error messages for the volume copy task.

**CKZ02131I PRE-COPIED VOLUME PAIRS
 ACCEPTED: SOURCE TARGET**

Explanation:
The DATA-MOVER program was NONE. The source and target volume serials have been paired as entered.

User response:
No action is required.

**CKZ02138I UNEXPECTED RESULTS FROM
 FCQUERY FQMAP; VOLSER= volser**

Explanation:
An ANTRQST FCQUERY FQMAP call returned unexpected results.

User response:
Please report this message to IBM Software Support.

**CKZ02140E UNEXPECTED CONDITION; error
 text**

Explanation:
An unexpected condition occurred while dispatching a copy task or while waiting for the completion of a copy task. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ02141E INTERNAL ERROR; error_text

Explanation:
This is an internal error. *error_text* has a description of the problem. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02142E MORE SOURCE VOLSERS THAN
 TARGET VOLSERS**

Explanation:
Volume pairing detected more source volumes than target volumes. Processing terminates.

User response:
This may have been caused by the use of TARGET-VOLS-SHOULD-BE-EMPTY eliminating target volumes. If unable to determine the cause, contact IBM

Software Support. Have available the listing that contains this message.

CKZ02144E ANTRQST LEVEL NOT SUPPORTED

Explanation:

For FlashCopy support, the level must be greater than four (4). For SnapShot support, the level must be greater than one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response:

Check with your system programmer for upgrading the system.

CKZ02145E ANTMMAIN NOT ACTIVE

Explanation:

An ANTRQST request failed because the system task, ANTMMAIN, is not active. Processing terminates.

User response:

Start the system task, ANTMMAIN.

CKZ02146E ANTRQST DID NOT RETURN ANY INFORMATION; RESTART ANTMMAIN SYSTEM TASK

Explanation:

The ANTRQST REQUEST=FCQUERY did not receive information for a device. Processing terminates.

User response:

Restart the ANTMMAIN system task, or, apply IBM APAR OW47323.

CKZ02147E ANTRQST ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=lllll VOLSER=volser

Explanation:

An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02148E SOURCE VOLUME NOT ELIGIBLE FOR PROCESSING; VOLSER=volser

Explanation:

A request for SDVCINFO or SQRVDVCS indicated the indicated volume serial is not eligible for FlashCopy or SnapShot. 'reason text' indicates the system response. For a source volume, processing terminates.

User response:

Eliminate source volumes which are not eligible for FlashCopy or Snapshot.

CKZ02148I TARGET VOLUME NOT ELIGIBLE FOR PROCESSING; VOLSER=volser WILL BE SKIPPED. 'reason text'

Explanation:

A request for SDVCINFO or SQRVDVCS indicated the indicated volume serial is not eligible for FlashCopy or SnapShot. 'reason text' indicates the system response. The target volume is eliminated for pairing; processing continues.

User response:

No action is required.

CKZ02149E SOURCE VOLSER: volser ACTIVE IN COPY RELATIONSHIP

Explanation

An FCQUERY request indicated the volser is currently in a copy relationship. The FCQUERY response is printed. For a source volser, processing terminates.

The format of the FCQUERY response for FlashCopy V1 is: devn, ssid, lss, cca, cu, serial, status

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- status is the current status of the device:

XRC

Device is in XRC pair

PPRC

Device is in PPRC pair

CC

Device is in CC session

FC

Device is in FlashCopy (no background copy)

FC..xxx%

Device is in FlashCopy (background copy)

SIMPLEX

Device is not in any copy status

The format of the FCQUERY response for FlashCopy V2 is:

devn, ssid, lss, cca, cu, serial, act, max, xc, pc, cc

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number

- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- act is the current number of FlashCopy relationships that the device has. This value is indicated in decimal format.
- max is the maximum number of FlashCopy relationships that the device may have. This value is indicated in decimal format.
- xc is either S indicating that the device is an XRC source volume, or is N indicating that the device is not an XRC source volume
- pc is either P indicating that the device is a PPRC primary volume, is S indicating that the device is an PPRC secondary volume, or is N for neither
- cc is either S indicating that the device is a concurrent copy source volume, or is N indicating that the device is not a concurrent copy source volume

User response:

Exclude that source volume from processing, or, ensure that all copy relationships have terminated before initiating the COPY.

CKZ02149I TARGET VOLSER: volser ACTIVE IN COPY RELATIONSHIP

Explanation

An FCQUERY request indicated the volser is currently in a copy relationship. The FCQUERY response is printed. For a target volser, processing continues, but that target volser will not be used by Db2 Cloning Tool.

The format of the FCQUERY response for FlashCopy V1 is: devn, ssid, lss, cca, cu, serial, status

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- status is the current status of the device:

XRC

Device is in XRC pair

PPRC

Device is in PPRC pair

CC

Device is in CC session

FC

Device is in FlashCopy (no background copy)

FC..xxx%

Device is in FlashCopy (background copy)

SIMPLEX

Device is not in any copy status

The format of the FCQUERY response for FlashCopy V2 is:

devn, ssid, lss, cca, cu, serial, act, max, xc, pc, cc

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- act is the current number of FlashCopy relationships that the device has. This value is indicated in decimal format.
- max is the maximum number of FlashCopy relationships that the device may have. This value is indicated in decimal format.
- xc is either S indicating that the device is an XRC source volume, or is N indicating that the device is not an XRC source volume
- pc is either P indicating that the device is a PPRC primary volume, is S indicating that the device is an PPRC secondary volume, or is N for neither
- cc is either S indicating that the device is a concurrent copy source volume, or is N indicating that the device is not a concurrent copy source volume

User response:

Exclude that source volume from processing, or, ensure that all copy relationships have terminated before initiating the COPY.

CKZ02150E NO INFORMATION RETURNED FOR SOURCE VOLSER=volser

Explanation:

SDVCINFO did not return information for the indicated source volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02151E NO TARGETS RETURNED FOR SOURCE VOLSER=volser

Explanation:

SQRYDVCS did not return any target volsers for the indicated source volume. Processing terminates.

User response:

Ensure that for every source volume, there is a target volume that meets the criteria required for FlashCopy or Snapshot. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02152E NO TARGET VOLSER FOUND FOR SOURCE VOLSER=*volser* LOC=*location additional_text*

Explanation

An attempt to find a target volume for the source volume failed. All of the target volumes are printed after this message, along with the reason for the failure to match with a source volume.

No suitable target volume was found for a COPY FULL based on the source volume's device type, FlashCopy capability, LSS, number of tracks, and sequence number (if KEEP-VOLUMES-SEQUENCE(Y) is specified). The *location* value indicates when the error appeared and is needed for IBM Software Support. Possible values for *location* are the following:

- **QFV01:** After the primary pairing stage, no target volumes were found for this source volume that have the required device type for copy by fast replication.
- **QFV02:** After the secondary pairing stage, no target volumes were found for this source volume that have the required device type for copy by fast replication
- **NFC01:** No target volumes were found for this source volume that have the required device type for slow copy replication.

The *additional_text* message string contains the following information:

```
[SOURCE | TARGET] volser TYPE=type MAN=man
FC=status_fc LSS=lss TRKS=trks SSER=sSER
type_of_sequence SEQ#= sequence_number
```

where:

SOURCE | TARGET

Identifies whether the volume is a source or target of the pairing.

volser

Source or target volume serial.

type

Device type.

man

Device's manufacturer (this information is output only for the fast replication method).

status_fc

Fast replication status for this volume. One of the following: AVAILABLE (device is available); IN COPY REL (device is in copy relation state); NOT ELIGIBLE (not eligible for fast copy relationship).

This information is output only for the fast replication method.

lss

LSS number. This information is output only for the fast replication method.

trks

The size of the volume in tracks.

sSER

Subsystem serial number. This information is output only for the fast replication method.

type_of_sequence

Type of volume sequence number. This value indicates from where the sequence number was generated and is one of the following:

VOLUME: FROM-VOLSER or FROM-VOLSER-DDN for source volumes or TO-VOLSER or TO-VOLSER-DDN for target volumes.

STOGRP: For source volumes, FROM-STORAGEGROUP, or FROM-USER-STORAGEGROUP if TO-STORAGEGROUP was used to retrieve target volumes. For target volumes, TO-STORAGEGROUP, or TO-USER-STORAGEGROUP if FROM-STORAGEGROUP was used to retrieve source volumes.

USRSGR: FROM-USER-STORAGEGROUP for source volumes or TO-USER-STORAGEGROUP for target volumes.

sequence_number

The volume sequence number (starting with 0) generated using the FROM parameter for source volumes or the TO parameter for target volumes.

The *additional_text* message string also provides information about why the pairing was not successful. For each target volume, one of the following reasons that a pairing was not successful is displayed:

tgt_volser PAIRED WITH src_volser

The target volume *tgt_volser* is already paired with another source volume *src_volser*.

tgt_volser NOT EMPTY

The target volume *tgt_volser* is not empty.

tgt_volser NOT FLASHCOPY CAPABLE/ELIGIBLE

The volume *tgt_volser* is not FlashCopy eligible.

tgt_volser FEWER TRACKS

The target volume is smaller (has fewer tracks) than the corresponding source volume.

tgt_volser DIFFERENT SEQUENCE NUMBERS

KEEP-VOLUMES-SEQUENCE(Y) was defined, but the target volume *tgt_volser* and the source volume have different volume sequence numbers.

tgt_volser DIFFERENT STORAGE CONTROLLER

The target volume *tgt_volser* and the source volume are on different storage controllers.

tgt_volser DIFFERENT TYPE

The source and target volumes have different types of devices.

tgt_volser UNEXPECTED CONDITION

An unexpected condition prevents the pairing of the source volume to the target volume *tgt_volser*.

tgt_volser IS SPACE EFFICIENT

The volume is space efficient.

tgt_volser FAILOVER/FAILBACK IN PROCESS

A failover or failback process is in progress.

tgt_volser IS CASCADED PPRC VOLUME

The volume *tgt_volser* is a cascaded PPRC volume.

tgt_volser UNKNOWN PROBLEM | UNKNOWN XTNT PROBLEM | UNKNOWN PM PROBLEM | UNKNOWN PM XTNT PROBLEM X'reason_code'**additional_info**

An unknown problem with reason code *reason_code* occurred. Additional information is provided in *additional_info*.

Db2 Cloning Tool_text FlashCopy_reason

For messages that are output when the target volume is not FlashCopy eligible, the reason text that is returned from the AOM QUERY FlashCopy is printed after the Db2 Cloning Tool rejection reason text. See [AOM QUERY FlashCopy return and reason codes](#) for additional information about the FlashCopy messages.

User response:

Ensure that for every source volume, there is a target volume that meets the criteria required. If VOLSER masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02153E ERROR DURING VVAS; RC=X'nnnn' REASON=X'nnnn' VOLSER=volser

Explanation:

An error occurred using VVAS to check a target volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02154I TARGET VOLUME ELIMINATED, VOLSER=volser | DATA SET OTHER THAN SYS1.VTOCIX OR SYS1.VVDS FOUND: datasetname

Explanation:

An unexpected data set was found on a target volume in response to TARGET-VOLS-SHOULD-BE-EMPTY. Processing continues.

User response:

No action is required, unless this causes fewer target volumes than source volumes for the pairing process.

CKZ02155I SMS SOURCE VOLSER=volser HAS BEEN PAIRED TO NON SMS TARGET VOLSER=volser | NON SMS SOURCE VOLSER=volser HAS BEEN PAIRED TO SMS TARGET VOLSER=volser

Explanation:

A source volume has been paired to a target volume that does not have the same SMS attribute. There may be problems when accessing or deleting data sets on the target volume.

User response:

No action is required.

CKZ02156E SOURCE VOLSER: volser NOT FLASHCOPY CAPABLE

Explanation:

An FCQUERY request indicated the volser is not FlashCopy capable. Information returned by FCQUERY is printed. For a source volser, processing terminates.

User response:

Exclude that source volume from processing, or, determine why the volume is not FlashCopy capable and correct if possible.

CKZ02156I TARGET VOLSER: volser NOT FLASHCOPY CAPABLE

Explanation:

An FCQUERY request indicated the volser is not FlashCopy capable. Information returned by FCQUERY is printed. For a target volser, processing continues, but that target volser will not be used.

User response:

Exclude that source volume from processing, or, determine why the volume is not FlashCopy capable and correct if possible.

CKZ02158I SLOW COPY WILL BE USED FROM SOURCE VOLSER= volser TO TARGET VOLSER= volser

Explanation:

FASTREP(PREF) has been specified and the copy between the two indicated volumes will be a slow copy rather than a fast FlashCopy or Snapshot.

User response:

No action is required.

CKZ02160E THE VOLSER SPECIFIED FOR A SOURCE USERCATALOG IS NOT A SOURCE VOLSER VOLSER=volser BCS=catalogname

Explanation:

The volser specified for a source catalog in the USERCATALOGS keyword is not a source volser. Processing terminates.

User response:

Correct the volser specified for the source catalog in the USERCATALOGS keyword to be source volser.

CKZ02161I SOURCE USERCATALOG WILL BE READ FROM TARGET VOLUME; VOLSER=volser BCS=catalogname

Explanation:

The source catalog will be read from the shown target volume.

User response:

No action is required.

CKZ02162I THE USERCATALOGS PAIRING DOES NOT MATCH THE VOLUME PAIRING; WILL USE THE VOLUME PAIRING

Explanation:

The source and target volumes that have catalogs on them are paired by the USERCATALOGS option, but they are not paired by volumes. They will be paired by volumes.

User response:

No action is required.

CKZ02163E INCREMENTAL FLASHCOPY RELATIONSHIP VOLUMES *srcvol* AND *tgtvol* CANNOT BE PAIRED. REASON: *reason*

Explanation

The source volume *srcvol* has an incremental FlashCopy relationship with the target volume *tgtvol*, but this pair does not meet the specified conditions of pairing volumes for one of the following reasons:

- The target volume *tgtvol* is an XRC primary volume, but the keyword FCTOXRCPRIARY is not specified.
- The target volume *tgtvol* is a PPRC primary volume, but the keyword FCTOPPRCPRIARY is not specified.
- The target volume *tgtvol* is an XRC primary volume and a PPRC primary volume, but one of the keywords FCTOXRCPRIARY or FCTOPPRCPRIARY is not specified.
- The option KEEP-VOLUMES-SEQUENCE(Y) is specified, and the volumes *srcvol* and *tgtvol* cannot be paired, because they have different sequence numbers.

User response:

Verify and correct either of the COPY command options for pairing volumes, or withdraw the existing

relationship of the source and target volumes and re-issue the COPY command.

CKZ02166E NO SOURCE VOLSER WITH CATALOG FOUND FOR TARGET VOLSER: *volser additional_text*

Explanation

An attempt to find a source volume that has a catalog on it for the target volume that has a catalog failed. All of the source volumes that were rejected are printed after this message.

No suitable target volume was found for a COPY FULL based on the source volume's device type, FlashCopy capability, LSS, number of tracks, and sequence number (if KEEP-VOLUMES-SEQUENCE(Y) is specified).

The *additional_text* message string contains the following information:

- For target volume data:

```
volser TYPE=type MAN=man SSER=sser TRKS=trks
type_of_sequence SEQ#= sequence_number
CATALOG=catalog
```

- For source volume data:

```
volser TYPE=type MAN=man SSER=sser TRKS=trks
type_of_sequence SEQ#= sequence_number
CATALOG=catalog reason_for_rejection
```

where:

volser

Source or target volume serial.

type

Device type.

man

Device's manufacturer (this information is output only for the fast replication method).

sser

Subsystem serial number. This information is output only for the fast replication method.

trks

The size of the volume in tracks.

type_of_sequence

Type of volume sequence number. This value indicates from where the sequence number was generated and is one of the following:

VOLUME: TO-VOLSER or TO-VOLSER-DDN for target volumes or FROM-VOLSER or FROM-VOLSER-DDN for source volumes.

STOGRP: For target volumes, TO-STORAGEGROUP, or TO-USER-STORAGEGROUP if FROM-STORAGEGROUP was used to retrieve source volumes. For source volumes, FROM-

STORAGEGROUP, or FROM-USER-STORAGEGROUP if TO-STORAGEGROUP was used to retrieve target volumes.

USRSGR: TO-USER-STORAGEGROUP for target volumes or FROM-USER-STORAGEGROUP for source volumes.

sequence_number

The volume sequence number (starting with 0) generated using the TO parameter for target volumes or the FROM parameter for source volumes.

catalog

The catalog that is on source or target volume.

reason_for_rejection

Can be one of the following:

PAIRED WITH *tgt_volser*: The volume is already paired with another target volume *tgt_volser*.

FEWER TRACKS: The target volume is smaller than the corresponding source volume

DIFFERENT SEQUENCE NUMBERS: The source volume has a different volume sequence number than the target volume, and KEEP-VOLUMES-SEQUENCE(Y) was specified.

DIFFERENT TYPE: The source and target volumes are different types of devices.

PERHAPS NO FASTREP RETAINED: The source and target volumes might not be capable of fast replication.

UNEXPECTED CONDITION: An unknown condition was encountered.

User response:

Ensure that for every target volume that has a catalog on it, there is a source volume with catalog on it that meets the required criteria. If VOLSER masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02201I **hh:mm:ss BCS BACKUP STARTED - PROGRAM REV=rrr | hh:mm:ss BCS BACKUP COMPLETED; RETURN CODE=nnn**

Explanation:

BCS backup processing message.

User response:

No action is required.

CKZ02205E **ALLOCATION FAILED FOR DSN: datasetname | DEALLOCATION FAILED FOR DSN: datasetname**

Explanation:

Dynamic allocation for a data set failed, or, dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ02206E **ERROR CALLING CKZ01VV1 ttttttt FUNCTION: function R15=nnnn RO=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. ttttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ02207W **ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ02209E **ERROR ACCESSING JOURNAL FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ02211E **JOURNAL CONTROL RECORD NOT FOUND | JOURNAL USER CATALOG RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02212E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL UCAT PAIR RECORD IS WRONG VERSION

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ02213E COUNT MISMATCH; RECORDS READ=rrrr PARM COUNT=pppp

Explanation:

The number of User Catalog records read from the Db2 Cloning Tool journal, rrrr, is not the same as the number indicated in the Db2 Cloning Tool journal control record, pppp. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02230I BCS BACKUP TASK COMPLETED; RETURN CODE=nn SYSOUT DD=ddname

Explanation:

A BCS backup task has ended.

User response:

No action is required if the RETURN CODE is zero. If the RETURN CODE is not zero, check the indicated sysout file for warning or error messages for the BCS backup task.

CKZ02240E UNEXPECTED CONDITION; error text

Explanation:

An unexpected condition occurred while dispatching a BCS backup task or while waiting for the completion of a BCS backup task. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02241I BCS=bcs dsname WILL BE BACKED UP TO DSN=backup dsname

Explanation:

The indicated BCS will be backed up to the indicated backup data set name. The backup data set name was derived from the CATWORK-DSN parameter.

User response:

No action is required.

CKZ02242E ERROR PARSING CATWORK-DSN: catwork dsn

Explanation:

An internal error occurred handling the CATWORK-DSN.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02301I hh:mm:ss EMC VOLUME COPY STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss EMC VOLUME COPY COMPLETED; RETURN CODE=nnn**

Explanation:

EMC VOLUME PAIRING processing message.

User response:

No action is required.

CKZ02304E OPEN FAILED FOR DDNAME=ddname

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ02307W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ02308E UNABLE TO LOAD PROGRAM: program name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ02309E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02310E DUPLICATE JOURNAL ENTRY;
 LOC=lllll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02311E JOURNAL VOLUME PAIR
 RECORD(S) NOT FOUND |
 JOURNAL UCAT PAIR RECORD(S)
 NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02312E JOURNAL UCAT PAIR RECORD IS
 WRONG VERSION**

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02330I EMCSNAPI COMPLETED; RETURN
 CODE=*return_code***

Explanation:

An EMCSNAPI call ended.

User response:

If the return code is 0, no action is required. If the return code is not 0, check the SYSOUT file for warning or error messages that are related to the volume snap. A return code of 4 may result if any target volume is online to another system.

**CKZ02340E VOLP ENTRY NOT FOUND FOR
 VOLSER=*volser***

Explanation:

An unexpected condition occurred while processing. A previously found VOLP entry cannot be found. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02341E INTERNAL ERROR; *reason*

Explanation:

An internal error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02342E MORE SOURCE VOLSERS THAN
 TARGET VOLSERS**

Explanation:

Volume pairing detected more source volumes than target volumes. Processing terminates.

User response:

This may have been caused by the use of TARGET-VOLS-SHOULD-BE-EMPTY eliminating target volumes or by some of the target volumes not having the necessary capabilities. If unable to determine the cause, contact IBM Software Support. Have available the listing that contains this message.

CKZ02343I *module-name* VERSION *version*

Explanation:

This message reports the version information retrieved from the indicated module.

User response:

No action is required.

CKZ02345E EMCSCF NOT ACTIVE; LOC=llll

Explanation:

An EMC request failed because the system task, EMCSCF, is not active. Processing terminates.

User response:

Start the system task, EMCSCF.

**CKZ02346E *module-name* ERROR; *request*
 type R15=*nnnn* X'*hhhh*' LOC=llll
 VOLSER=*volser***

Explanation:

The call to the indicated module for the indicated request type failed. The return code is displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02347E *module-name* ERROR; *request*
 type RETURN CODE=*nnnn* X'*hhhh*'
 REASON CODE=*nnnn* X'*hhhh*'
 LOC=lllll VOLSER=*volser***

Explanation:

The call to the indicated module for the indicated request type failed. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ02348E SOURCE VOLUME NOT ELIGIBLE FOR PROCESSING; VOLSER=volser

Explanation:

It has been determined that the indicated volume serial is not eligible for use. 'reason text' indicates the reason. Processing terminates.

User response:

Eliminate source volumes which are not eligible for use.

CKZ02348I TARGET VOLUME NOT ELIGIBLE FOR PROCESSING; VOLSER=volser WILL BE SKIPPED. 'reason text'

Explanation:

It has been determined that the indicated volume serial is not eligible for use. 'reason text' indicates the reason. Processing continues.

User response:

No action is required.

CKZ02352E NO TARGET VOLSER FOUND FOR SOURCE VOLSER=volser LOC=location additional_text

Explanation

An attempt to find a target volume for the source volume failed. All of the target volumes are printed after this message, along with the reason for the failure to match with a source volume.

No suitable target volume was found for a COPY FULL based on the source volume's device type, EMC SNAP capability, number of tracks, and sequence number (if KEEP-VOLUMES-SEQUENCE(Y) is specified). The *location* value indicates when the error appeared and is needed for IBM Software Support.

The *additional_text* message string contains the following information:

- For source volume data:

```
volser TYPE=type MAN=man SSER=sser TRKS=trks
type_of_sequence SEQ#= sequence_number
```

- For target volume data:

```
volser TYPE=type MAN=man SSER=sser TRKS=trks
type_of_sequence SEQ#= sequence_number
reason_for_rejection
```

where:

volser

Source or target volume serial.

type

Device type.

man

Device's manufacturer (this information is output only for the fast replication method).

sser

Subsystem serial number. This information is output only for the fast replication method.

trks

The size of the volume in tracks.

type_of_sequence

Type of volume sequence number. This value indicates from where the sequence number was generated and is one of the following:

VOLUME: FROM-VOLSER or FROM-VOLSER-DDN for source volumes or TO-VOLSER or TO-VOLSER-DDN for target volumes.

STOGRP: For target volumes, TO-STORAGEGROUP, or TO-USER-STORAGEGROUP if FROM-STORAGEGROUP was used to retrieve source volumes. For source volumes, FROM-STORAGEGROUP, or FROM-USER-STORAGEGROUP if TO-STORAGEGROUP was used to retrieve target volumes.

USRSGR: FROM-USER-STORAGEGROUP for source volumes or TO-USER-STORAGEGROUP for target volumes.

sequence_number

The volume sequence number (starting with 0) generated using the FROM parameter for source volumes or the TO parameter for target volumes.

reason_for_rejection

Can be one of the following:

PAIRED WITH *src_volser*: The target volume is already paired with another source volume *src_volser*.

FEWER TRACKS: The target volume is smaller than the corresponding source volume.

DIFFERENT SEQUENCE NUMBERS: The source volume has a different volume sequence number than the target volume, and KEEP-VOLUMES-SEQUENCE(Y) was specified.

DIFFERENT TYPE: The source and target volumes are different types of devices.

NOT SNAP ELIGIBLE: The volume is not eligible for SNAP operation.

DIFFERENT STORAGE CONTROLLER: The target volume and the source volume are on different storage controllers.

UNEXPECTED CONDITION: An unknown condition was encountered.

User response:

Ensure that for every source volume, there is a target volume that meets the required criteria. If VOLSER masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

**CKZ02353E ERROR DURING VVAS; RC=X'nnnn'
REASON=X'nnnn' VOLSER=volser**

Explanation:

An error occurred using VVAS to check a target volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02354I TARGET VOLUME ELIMINATED,
VOLSER=volser | DATA SET OTHER
THAN SYS1.VTOCIX OR
SYS1.VVDS FOUND: datasetname**

Explanation:

An unexpected data set was found on a target volume in response to TARGET-VOLS-SHOULD-BE-EMPTY. Processing continues.

User response:

No action is required, unless this causes fewer target volumes than source volumes for the pairing process.

**CKZ02355I SMS SOURCE VOLSER=volser HAS
BEEN PAIRED TO NON SMS
TARGET VOLSER=volser | NON
SMS SOURCE VOLSER=volser HAS
BEEN PAIRED TO SMS TARGET
VOLSER=volser**

Explanation:

A source volume has been paired to a target volume that does not have the same SMS attribute. There may be problems when accessing or deleting data sets on the target volume.

User response:

No action is required.

**CKZ02356I CURRENT SNAP RELATIONSHIPS
FOR type VOLSER: volser volser
relationship information**

Explanation:

The current snap relationships for the volser are listed. 'type' indicates if the volser is a SOURCE or TARGET volume.

User response:

No action is required.

**CKZ02360E THE VOLSER SPECIFIED FOR A
SOURCE USERCATALOG IS NOT A
SOURCE VOLSER VOLSER=volser
BCS=catalogname**

Explanation:

The volser specified for a source catalog in the USERCATALOGS keyword is not a source volser. Processing terminates.

User response:

Correct the volser specified for the source catalog in the USERCATALOGS keyword to be source volser.

**CKZ02361I SOURCE USERCATALOG WILL BE
READ FROM TARGET VOLUME;
VOLSER=volser BCS=catalogname**

Explanation:

The source catalog will be read from the shown target volume.

User response:

No action is required.

**CKZ02364E SOURCE VOLUME: src_volser
WITHOUT CATALOG IS PAIRED BY
DIFFERENTIAL TO TARGET
VOLUME: tgt_volser WITH
CATALOG**

Explanation:

The source volume *src_volser* does not have a catalog and has a DIFFERENTIAL SNAP to the target volume *tgt_volser*, which does have a catalog. Processing terminates.

User response:

Wait for the current session to complete, then clean up and re-run the COPY command. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

**CKZ02365E SOURCE VOLUME: src_volser WITH
CATALOG PAIRED WITH volser1
AND PAIRED WITH volser2
TARGET VOLUMES**

Explanation:

The source volume *src_volser* has a catalog on it and has a DIFFERENTIAL SNAP to two target volumes. Processing terminates.

User response:

Wait for the current session to complete, then clean up and re-run the COPY command. If unable to determine

the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02366E NO SOURCE VOLSER WITH CATALOG FOUND FOR TARGET VOLSER: *volser additional_text*

Explanation

An attempt to find a source volume that has a catalog on it for the target volume that has a catalog failed. All of the source volumes that were rejected are printed after this message.

No suitable source volume was found for a SNAP VOLUME based on the target volume's device type, EMC SNAP capability, storage subsystem, number of tracks, and volume sequence number (if KEEP-VOLUMES-SEQUENCE(Y) is specified).

The *additional_text* message string contains the following information:

- For target volume data:

```
volser TYPE=type MAN=man SSER=sser TRKS=trks
type_of_sequence SEQ#= sequence_number
CATALOG=catalog
```

- For source volume data:

```
volser TYPE=type MAN=man SSER=sser TRKS=trks
type_of_sequence SEQ#= sequence_number
CATALOG=catalog reason_for_rejection
```

where:

volser

Source or target volume serial.

type

Device type.

man

Device's manufacturer (this information is output only for the fast replication method).

sser

Subsystem serial number. This information is output only for the fast replication method.

trks

The size of the volume in tracks.

type_of_sequence

Type of volume sequence number. This value indicates from where the sequence number was generated and is one of the following:

VOLUME: TO-VOLSER or TO-VOLSER-DDN for target volumes or FROM-VOLSER or FROM-VOLSER-DDN for source volumes.

STOGRP: For target volumes, TO-STORAGEGROUP, or TO-USER-STORAGEGROUP if FROM-STORAGEGROUP was used to retrieve source

volumes. For source volumes, FROM-STORAGEGROUP, or FROM-USER-STORAGEGROUP if TO-STORAGEGROUP was used to retrieve target volumes.

USRSGR: TO-USER-STORAGEGROUP for target volumes or FROM-USER-STORAGEGROUP for source volumes.

sequence_number

The volume sequence number (starting with 0) generated using the TO parameter for target volumes or the FROM parameter for source volumes.

catalog

The catalog that is on source or target volume.

reason_for_rejection

Can be one of the following:

PAIRED WITH *tgt_volser*: The source volume is already paired with another target volume *tgt_volser*.

FEWER TRACKS: The target volume is smaller than the corresponding source volume.

DIFFERENT SEQUENCE NUMBERS: The source volume has a different volume sequence number than the target volume, and KEEP-VOLUMES-SEQUENCE(Y) was specified.

DIFFERENT TYPE: The source and target volumes are different types of devices.

NOT SNAP ELIGIBLE: The volume is not SNAP eligible.

DIFFERENT STORAGE CONTROLLER: The target volume and the source volume are on different storage controllers.

UNEXPECTED CONDITION: An unknown condition was encountered.

User response:

Ensure that for every target volume that has a catalog on it, there is a source volume with catalog on it that meets the required criteria. If VOLSER masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02401I hh:mm:ss VOLUME RECORDS SORTER STARTED - PROGRAM REV=*revision* | hh:mm:ss VOLUME RECORDS SORTER COMPLETED; RETURN CODE=*return_code*

Explanation:

This informational message is a volume records sorter processing message.

User response:
No action is required.

CKZ02402I OPERATION: *operation* STARTED

Explanation

This informational message is an operations processing message. *operation* is one of the following:

- SORT OF JOURNAL VOLUME PAIR RECORDS BY SOURCE FIELDS
- SORT OF JOURNAL VOLUME PAIR RECORDS BY TARGET FIELDS
- SORT VOLUME WITH CATALOGS RECORDS
- SORT SOURCE DATA RECORD POINTER
- SORT TARGET VOLUME RECORDS
- SORT VOLUME WITH CATALOGS RECORDS FOR RESTORE-FROM-DUMPTAPES
- LINK VOLUME WITH CAT.RECORDS TO JOURNAL VOLUME PAIR RECORDS

User response:
No action is required.

CKZ02403I OPERATION: *operation* COMPLETED SUCCESSFULLY

Explanation

This informational message is an operations processing message. *operation* is one of the following:

- SORT OF JOURNAL VOLUME PAIR RECORDS BY SOURCE FIELDS
- SORT OF JOURNAL VOLUME PAIR RECORDS BY TARGET FIELDS
- SORT VOLUME WITH CATALOGS RECORDS
- SORT SOURCE DATA RECORD POINTER
- SORT TARGET VOLUME RECORDS
- SORT VOLUME WITH CATALOGS RECORDS FOR RESTORE-FROM-DUMPTAPES
- LINK VOLUME WITH CAT.RECORDS TO JOURNAL VOLUME PAIR RECORDS

User response:
No action is required.

CKZ02404E OPERATION: *operation* FAILED. REASON: *reason*

Explanation:

The operation that is listed in the message completed with error, for the reason *reason* that is listed in the message. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ02407W ERROR CALLING CKZ01HEX; FUNCTION: *function* R15=*register15*

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ02601I *hh:mm:ss* XRC PAIRS DATA CONTAINER STARTED - PROGRAM REV=*revision* COMMAND: INIT | *hh:mm:ss* XRC PAIRS DATA CONTAINER COMPLETED. COMMAND: INIT

Explanation:

This informational message is an XRC pairs data container processing message.

User response:
No action is required.

CKZ02604E WRONG COMMAND. COMMAND: *command* VV2 AREA ADDRESS: *value*

Explanation:

An incorrect command was sent to the XRC pairs data container. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ02607W ERROR CALLING CKZ01HEX; FUNCTION: *function* R15=*return_code*

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ02608E UNABLE TO LOAD PROGRAM: *program_name*

Explanation:

The indicated *program_name* was not found. Processing terminates.

User response:
Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ02610I INFORMATION ABOUT XRC SOURCE VOLUMES

Explanation:

Information about the XRC source volumes was retrieved.

User response:

No action is required.

CKZ02611I NO INFORMATION ABOUT XRC SOURCE VOLUMES
Explanation:

Information about the XRC source volumes was not found.

User response:

No action is required.

**CKZ02626E ERROR CALLING CKZ01VV2
table_name FUNCTION: function
R15=return_code R0=reason_code
LOC=location**
Explanation:

A problem occurred using a dataspace. *table_name* is the name of the internal table. *location* is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages and the CKZINI member.

CKZ02645E ANTMAIN NOT ACTIVE
Explanation:

An ANTRQST request failed because the ANTMAIN system task is not active. Processing terminates.

User response:

Start the ANTMAIN system task.

**CKZ02647E ANTRQST ERROR; request_type
RETURN CODE=return_code
return_code_in_hex REASON
CODE=reason_code
reason_code_in_hex LOC=location
VOLSER=volser**
Explanation:

An ANTRQST request failed. *request_type* indicates if the macro was incorrect, or if the data mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. *location* is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02920W MESSAGE NUMBER NOT FOUND;
nnnn x'nnnn'**
Explanation:

The message number for an ANTRQST error was not found in an internal messages table. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ03001I hh:mm:ss COPY STARTED -
PROGRAM REV=rrr | hh:mm:ss
COPY COMPLETED; RETURN
CODE=nnn**
Explanation:

COPY task processing message.

User response:

No action is required.

**CKZ03003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**
Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ03004E OPEN FAILED FOR
DDNAME=ddname**
Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ03005E ALLOCATION FAILED FOR DSN:
datasetname | ALLOCATION
FAILED FOR DDNAME: ddname**
Explanation:

Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ03005W DEALLOCATION FAILED FOR
DDNAME: ddname**
Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software

Support. Have available the listing containing these messages.

**CKZ03008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ03012E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn**

Explanation:

The program was not able to establish an estaex environment. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03021E ADRDSSU COPY FAILED;
R15=nnnn**

Explanation:

A non-zero return code was received from ADRDSSU. The ADRDSSU messages will be contained in the sysout for the copy task. Processing terminates.

User response:

Check the ADRDSSU messages for the cause of the failure. If assistance is required, contact IBM Software Support. Have available the listing that contains this message.

**CKZ03022E DATA MOVER SETTING IS
INVALID; VALUE= value**

Explanation:

An internal error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ03023E INTERNAL ERROR; reason

Explanation:

An internal error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03041E CKZ00900 UNEXPECTED
RESULTS; error text**

Explanation:

An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03042E NO MATCH FOUND FOR DSS TASK
NUMBER: nnn**

Explanation:

The DSS task number was not matched to any of the volume pairs for this task. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03043E NO MATCH FOUND FOR TASK DD
IN VOLUME PAIRS**

Explanation:

The copy task was dispatched, but, no volume pairs had been assigned to it. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ03099E ABEND DURING COPY COMMAND

Explanation:

An abend occurred for a copy task. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03101I hh:mm:ss CLIP TARGETS STARTED
- PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss CLIP
TARGETS COMPLETED; RETURN
CODE=nnn**

Explanation:

Offline target processing message.

User response:

No action is required.

**CKZ03103I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ03104E OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ03105E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ03105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ03108E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ03131E CAPTURE FOR UCB FAILED; UCB
ADDRESS: nnnnnnnn R15: nnnn
TARGET VOLUME SERIAL: volser**

Explanation:

An attempt to use IOSCAPU to capture a UCB failed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03132E UNCAPTURE FOR UCB FAILED;
UCB ADDRESS: nnnnnnnn R15:
nnnn TARGET VOLUME SERIAL:
volser**

Explanation:

An attempt to use IOSCAPU to uncapture a UCB failed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03133E EXCP FAILED FOR DEVICE: nnnn
TARGET VOLSER: volser SYNAD
TEXT: text**

Explanation:

An error occurred reading the volume label for an offline target device. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03134E VOLUME SERIAL FOR DEVICE:
nnnn IS existing volser; VOLUME
SERIAL SHOULD BE: paired source
volser**

Explanation:

The volume serial number for the indicated device is 'existing volser'. Based on the volume pairs specified, the expected volume serial number is 'paired source volser'. Processing terminates.

User response:

Correct the volume pairs specified.

**CKZ03135E ICKDSF COMMAND FAILED FOR
TARGET VOLSER: volser**

Explanation:

The invocation of ICKDSF to change a device label failed. The messages from ICKDSF are printed.

User response:

If unable to determine the reason for the failure from the associated ICKDSF messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ03136W IEEVARYD VARY ONLINE FAILED
FOR TARGET VOLSER: volser
INVALID PARAMETERS**

Explanation:

The parameters given to IEEVARYD are incorrect. Processing continues.

User response:

Contact IBM Software Support. Have available the listing that contains this message. The indicated target volume serial will need to be manually varied online in order to proceed with Db2 Cloning Tool RENAME processing.

**CKZ03137W IEEVARYD VARY ONLINE FAILED
FOR TARGET VOLSER: volser
DEVICE: nnnn RETURN CODE:
nnnnnnnn REASON CODE:
nnnnnnnn**

Explanation:

The vary online for the indicated device failed. Processing continues.

User response:

When the problem that caused the vary to fail is corrected, the indicated target volume serial will need to be manually varied online in order to proceed with Db2 Cloning Tool RENAME processing.

CKZ03138W IEEVARYD VARY ONLINE FAILED FOR TARGET VOLSER: volser DEVICE: nnnn R15: nnnnnnnn

Explanation:

The vary online for the indicated device failed. Processing continues.

User response:

When the problem that caused the vary to fail is corrected, the indicated target volume serial will need to be manually varied online in order to proceed with Db2 Cloning Tool RENAME processing.

CKZ03140I DEVICE NUMBER: nnnn SUCCESSFULLY CHANGED TO VOLUME SERIAL: volser

Explanation:

The label of the indicated device has been changed by ICKDSF to the indicated volume serial number.

User response:

No action is required.

CKZ03141I VOLUME SERIAL: volser DEVICE NUMBER: nnnn IS NOW ONLINE

Explanation:

The indicated target volume is now online to the current image.

User response:

No action is required.

CKZ03142I DEVICE NUMBER: nnnn IS ONLINE AS TARGET VOLUME SERIAL: volser

Explanation:

The indicated device is currently online with the target volume serial.

User response:

No action is required.

CKZ03143E DEVICE NUMBER: nnnn IS ONLINE AS SOURCE VOLUME SERIAL: volser

Explanation:

The indicated device is currently online with the source volume serial. The device is expected to be either offline or online with the target volume serial. Processing terminates.

User response:

The device should be taken offline and the command rerun.

CKZ03144E DEVICE NUMBER: nnnn HAS UNEXPECTED VOLUME SERIAL: volser

Explanation:

The indicated device is currently online with a volume serial that is neither the source or target volume serial. The device is expected to have a volume serial that is either the source or target volume serial. Processing terminates.

User response:

Verify the device has not been overlaid with the contents of the wrong volume.

CKZ03145I DEVICE NUMBER: nnnn ALREADY CHANGED TO TARGET VOLUME SERIAL: volser

Explanation:

The indicated device currently has the desired target volume serial.

User response:

No action is required.

CKZ03146I DEVICE NUMBER: nnnn ICKDSF FAILED; DEVICE MAY HAVE ALREADY BEEN CLIPPED

Explanation:

The clip of the indicated device by ICKDSF has failed. This failure is probably caused by the device having already been clipped.

User response:

See the next Db2 Cloning Tool message in the listing to determine the appropriate action.

CKZ03148I VOLUME SERIAL: vvvvvv DEVICE NUMBER: nnnn NOT CLIPPED TO VOLUME SERIAL: vvvvvv DUE TO SIMULATION

Explanation:

The clip of the indicated device was not done because this run is a simulation.

User response:

No action is required.

CKZ03201I hh:mm:ss CATALOG FUNCTIONS STARTED - PROGRAM REV=*revision* | hh:mm:ss CATALOG FUNCTIONS COMPLETED; RETURN CODE=*return_code*

Explanation:

This informational message is a catalog functions processing message.

User response:

No action is required.

CKZ03206E **ERROR CALLING CKZ01VV1**
internal_table_name FUNCTION:
function R15=register15
R0=register0 LOC=location

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ03207W **ERROR CALLING CKZ01HEX;**
FUNCTION: function
R15=register15

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ03208E **UNABLE TO LOAD PROGRAM:**
program_name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ03220E **ERROR CALLING IGGCSI00;**
MODULE:X module REASON:
reason RETURN CODE: return_code
REG R15: register15

Explanation:

An error occurred calling the IGGCSI100 program module.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ03221W **ENTRY RETURNED BY IGGCSI00**
MARKED AS IN ERROR; MODULE:
X'module' REASON: reason
RETURN CODE: return_code

Explanation:

An entry was returned by IGGCSI00 marked as an error.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ03222E **ERROR CREATING COMMAND**
STRING; LOC=location

Explanation:

This is an internal error. A problem occurred in the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ03239E **INTERNAL ERROR; LOC=location**

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ03240I **COMMAND: command_text**

Explanation:

This informational messages displays the command text used by the CATALOG FUNCTIONS module.

User response:

No action is required.

CKZ03241I **COMMANDS WILL NOT BE**
EXECUTED DUE TO SIMULATION
MODE

Explanation:

The COPY command was run in SIMULATE mode. Therefore, the command in the CATALOG FUNCTIONS module is not run.

User response:

No action is required.

CKZ03501I **hh:mm:ss CHECK USERCATALOGS**
STARTED - PROGRAM REV=rrr |
hh:mm:ss CHECK USERCATALOGS
COMPLETED; RETURN CODE=nnn

Explanation:

CHECK USERCATALOGS processing message.

User response:

No action is required.

CKZ03503I **DDNAME=ddname ALLOCATED**
FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ03505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

No action is required.

**CKZ03507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ03559E ALLOCATION FOR USERCATALOG
FAILED; DSN=bcs dsname**

Explanation:

The BCS dsname was not successfully allocated for further checking.

User response:

Check that the user catalogs have been specified correctly.

**CKZ03560I WAITING FOR SHARED CONTROL
OF BCS bcs name**

Explanation:

The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job. Processing continues.

User response:

No action is required.

**CKZ03561E UNABLE TO ALLOCATE BCS: bcs
name; WAIT TIME LIMIT
EXCEEDED**

Explanation:

The wait for shared control of the BCS has exceeded the wait time limit. Processing terminates.

User response:

Change the scheduling of the jobs so the Db2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the Db2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

**CKZ03570E SECURITY PRODUCT DENIED
ACCESS TO DSN: datasetname**

Explanation:

The dsn indicated is not authorized for alter by your security product. If the RACF profile that is associated is returned, it will be displayed. Processing terminates.

User response:

Change the dsn to one you can use, or, have your security administrator give you 'ALTER' authority to the data set.

**CKZ03571E RACROUTE ERROR; SAF RC=nnnn
RACF RC=nnnn RACF REASON
CODE=nnnn**

Explanation:

An unexpected return code from SAF or RACF occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ04001I hh:mm:ss BCS BACKUP STARTED -
PROGRAM REV=rrr | hh:mm:ss
BCS BACKUP COMPLETED;
RETURN CODE=nnn**

Explanation:

BCS BACKUP processing message.

User response:

No action is required.

**CKZ04003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ04004E OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ04005E ALLOCATION FAILED FOR DSN:
datasetname | ALLOCATION
FAILED FOR DDNAME: ddname**

Explanation:

Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software

Support. Have available the listing containing these messages.

**CKZ04005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ04007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ04008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ04020W ERROR CALLING IGGCSI00;
MODULE: SMS_moduleID REASON:
SMS_reason_code RETURN CODE:
SMS_return_code REG R15:
return_code**

Explanation:

An error occurred calling the IGGCSI100 program module when the RLS status of the source catalog was being verified. The source catalog will be marked as non-RLS. Processing continues.

User response:

If you are unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ04021W ENTRY RETURNED BY IGGCSI00
MARKED AS IN ERROR; MODULE:
SMS_moduleID REASON:
SMS_reason_code RETURN CODE:
SMS_return_code**

Explanation:

An entry was returned by IGGCSI00 marked as an error when the RLS status of the source catalog was

being verified. The source catalog will be marked as non-RLS. Processing continues.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ04022W THE FOLLOWING ERROR FLAGS
WERE RETURNED BY IGGCSI00
FOR CATALOG catalog_name:
list_of_errors**

Explanation:

When the RLS status of the source catalog was being verified, IGGCSI00 returned one or more catalog error flags for source catalog *catalog_name*. *list_of_errors* is the list of error flags. The source catalog will be marked as non-RLS. Processing continues.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ04023W THE FOLLOWING ERROR FLAGS
WERE RETURNED BY IGGCSI00
FOR ENTRY catalog_name:
list_of_errors**

Explanation:

When the RLS status of the source catalog was being verified, IGGCSI00 returned one or more entry error flags for source catalog *catalog_name*. *list_of_errors* is the list of error flags. The source catalog will be marked as non-RLS. Processing continues.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ04024W IGGCSI00 DID NOT RETURN
ENOUGH DATA**

Explanation:

IGGCSI00 did not return enough data when the RLS status of the source catalog was being verified. The list of detected errors follows. The source catalog will be marked as non-RLS. Processing continues.

User response:

If you are unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ04028I THE CATALOG: catalog_name IS IN
RLS MODE**

Explanation:

The catalog *catalog_name* is in RLS mode.

User response:

No action is required.

CKZ04030E AMSOPEN FAILED; R15=nnnn

Explanation:

An attempt was made to issue an IDCAMS command. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ04031I NON-ZERO RETURNED BY
IDCAMS; RC=nnnn**

Explanation:

An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will continue if the IDCAMS return code is less than or equal to 4. Processing will terminate if the IDCAMS return code is greater than 4 or if IDCAMS returned more than 50 errors.

User response:

If the IDCAMS return code was greater than 8, there is a problem with the source user catalog that will probably prevent Db2 Cloning Tool backing it up. Correct the problem before invoking Db2 Cloning Tool COPY.

**CKZ04033I IDCAMS VERIFY FOR BCS WAS
SUCCESSFUL**

Explanation:

An IDCAMS VERIFY was issued for the BCS to be backed up. The VERIFY completed normally.

User response:

No action is required.

**CKZ04034I IDCAMS EXAMINE INDEXTST
FOR BCS WAS SUCCESSFUL**

Explanation:

An IDCAMS EXAMINE INDEXTST was issued for the BCS to be backed up. The EXAMINE completed normally.

User response:

No action is required.

**CKZ04035I NUMBER OF RECORDS READ
FROM BCS: nnnnnnnn**

Explanation:

The indicated number of records were read from the BCS.

User response:

No action is required.

**CKZ04036I NUMBER OF RECORDS WRITTEN
TO BACKUP: nnnnnnnn**

Explanation:

The indicated number of records were written to the BCS backup file.

User response:

No action is required.

**CKZ04037I IDCAMS VERIFY FOR RLS
CATALOG WAS SKIPPED**

Explanation:

The IDCAMS VERIFY was not called for the source catalog, because this catalog is in RLS mode.

User response:

No action is required.

**CKZ04038I IDCAMS EXAMINE INDEXTST
FOR RLS CATALOG WAS SKIPPED**

Explanation:

The IDCAMS EXAMINE INDEXTST was not called for the source catalog, because this catalog is in RLS mode.

User response:

No action is required.

**CKZ04040W BCS RECORD HAS BEEN SKIPPED;
LENGTH MISMATCH; VSAM
RECORD LENGTH: nnnnnnnn BCS
RECORD LENGTH: nnnnnnnn**

Explanation:

There is a mismatch in records lengths. VSAM READ returned a length that was different than the length indicated in the BCS record. The BCS record is printed, but, because of this length error, the BCS record is not written to the backup file.

User response:

No action is required. But, this does indicate a problem with the entry in the source usercatalog.

**CKZ04045E ERROR ACCESSING
BCS=*bcs_data_set_name*;
LOC=*location***

Explanation:

A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response

See the associated CKZERRnnE error messages. If the indicated BCS is in RLS mode, run the following MVS command:

```
F CATALOG,UNALLOCATE(bcs_data_set_name)
```

Then rerun the COPY step. If you are unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ04046E ERROR CALLING CKZ00045;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ00045 to read a source catalog. Processing terminates.

User response:

See associated CKZnnnnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ04047W READING OF SOURCE
USERCATALOG BYPASSED DUE TO
SIMULATION AND SOURCE
VOLUME BEING OFFLINE
VOLSER=volser BCS=catalogname**

Explanation:

The source catalog cannot be read because this is a simulation run and the source volume is not online. For a simulation run the source catalog is read from the source volume. Processing continues.

User response:

None, unless a RENAME simulation is desired. To have RENAME SIM process with this catalogs entries either bring the source volume online for COPY SIM or run COPY without SIM.

**CKZ04060I WAITING FOR EXCLUSIVE
CONTROL OF BCS bcs name**

Explanation:

The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response:

No action is required.

**CKZ04061E UNABLE TO ALLOCATE BCS: bcs
name; WAIT TIME LIMIT
EXCEEDED**

Explanation:

The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response:

Change the scheduling of the jobs so the Db2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the Db2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

**CKZ04070E ERROR DURING UCBLINK FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using the UCBLINK macro. Processing terminates.

User response:

If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ04501I CKZ00045 PROGRAM REV=x

Explanation:

Program CKZ00045 displaying its versioning information.

User response:

No action is required.

**CKZ04502E Requested catalog xxx not found
on volume vvv**

Explanation:

The specified catalog was not found on the volume. Processing terminates.

User response:

Correct USERCATALOGS parameters to specify the source volser where the catalog resides.

CKZ04513E VOLSER xxx not found online

Explanation:

VOLSER was not found on the system. Processing terminates.

User response:

Correct the JOB's volser and resubmit.

**CKZ04515E UNRECOGNIZED DEVICE TYPE:
devicetype**

Explanation:

The UCBTYP for to volume is not defined as DASD. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ04517E Error processing VVCNs: message

Explanation:

An error occurred during VVDS processing. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ04521E UCBLINK ERROR FOR
VOLSER=volser RETURN CODE=nn
REASON CODE=nn**

Explanation:

An error occurred during UCBLINK processing. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

User response:

Contact IBM Software Support.

**CKZ04629E VVR IS MISSING
volumeinformationcell
DSN=dsname**

Explanation:

The VSAM Volume Record for the specified DSN was not located in the VVDS. Processing terminates.

User response:

Ensure the validity of VVDS by executing a DIAGNOSE. If the problem persists contact IBM Software Support, for assistance.

**CKZ04631E UCBLOOK FAILED FOR VOLUME
volser, RC='returncode',
RSN='reasoncode'**

Explanation:

An error occurred will attempting to pin the UCB for the specified volume. Processing terminates.

User response:

Contact IBM Software Support.

**CKZ04632E DSN=dsname NOT FOUND -
module(returncode - reason code)**

Explanation:

An error occurred will attempting to retrieve volume data from the catalog. Processing terminates.

User response:

Contact IBM Software Support.

**CKZ04633A DEBLOCK ERROR(code) volser
CCHHR=cyl_head_record
RDF_OFFSET=offset**

Explanation:

An error occurred while deblocking a control interval. Processing continues, but certain affected logical records may be unrecoverable. The CCHHR value indicates the physical block that is in error on the DASD device. The physical contents of that track can be displayed using the ADRDSSU PRINT command. ADRDSSU command example: PRINT TRACKS (X'cccc',X'hh',X'cccc',X'hh') INDY(volser)

User response:

Contact IBM Software Support. Provide the sysout of the failing job, including the Joblog output. Tech Support may also request the ADRDSSU output for the PRINT TRACKS command.

**CKZ04634A Invalid CIDF Data X'????????'
volser CCHHR=cccc_hhhh_rr**

Explanation:

A control interval failed validation and could not be deblocked. Processing continues, but certain affected logical records may be unrecoverable. The CCHHR value indicates the physical block that is in error on the

DASD device. The physical contents of that track can be displayed using the ADRDSSU PRINT command. ADRDSSU command example: PRINT TRACKS (X'cccc',X'hh',X'cccc',X'hh') INDY(volser)

User response:

Contact IBM Software Support. Provide the sysout of the failing job, including the Joblog output. Tech Support may also request the ADRDSSU output for the PRINT TRACKS command.

**CKZ04635A SPANNED RECORD ERROR - volser
CCHHR=cccc_hhhh_rr**

Explanation:

While processing a spanned record, an error occurred. Spanned records, for all but ESDS, must be re-assembled by using the data an the index entry - specifically the sequence of segments is controlled by the FLP in the index entry. The Volser and CCHHR value represents the physical address of the beginning of the Control Interval that is in error. Processing continues.

User response:

Refer to other associated messages issued in conjunction with this error.

**CKZ04636W TRUNCATED RECORD -
RBA=X'xxxxxxxx.xxxxxxxx'**

Explanation:

While processing a spanned record, an error occurred. Processing continues.

User response:

Refer to other associated messages issued in conjunction with this error.

**CKZ04638E EXTENDED-FORMAT STRIPED
DATASETS NOT SUPPORTED.**

Explanation:

An attempt was made to process a striped object in EXCPMODE. This is not supported. Processing terminates.

User response:

Contact IBM Software Support.

**CKZ04639E EXTENDED-FORMAT
COMPRESSED DATASETS NOT
SUPPORTED.**

Explanation:

An attempt was made to process a compressed object in EXCPMODE. This is not supported. Processing terminates.

User response:

Contact IBM Software Support.

**CKZ04640E DATASET NOT FOUND OR NOT
SPECIFIED**

Explanation:

The EXCPMODE interface was incorrectly used. Processing terminates.

User response:

Contact IBM Software Support.

CKZ04641E INVALID XXSET CALL

Explanation:

The XXSET macro had one or more incorrectly specified arguments. Processing terminates.

User response:

Contact IBM Software Support.

CKZ04642E VOLSER NOT SPECIFIED

Explanation:

The XXSET macro had no VOLSER argument coded. Processing terminates.

User response:

Contact IBM Software Support.

CKZ04643E INVALID VOLSEQ VALUE SPECIFIED

Explanation:

The XXSET macro had an invalid SEQ argument coded. Processing terminates.

User response:

Contact IBM Software Support.

CKZ04644E DUPLICATE XXSET FOR component DSN=datasetname VOL=volser SEQ=sequence#

Explanation:

Multiple XXSET macros with the same arguments were processed. Processing terminates.

User response:

Contact IBM Software Support.

CKZ04645E First VOLSEQ Missing for Data Component DSN=comp.name

Explanation:

The first volume of a multi-volume set for the Data Component was not identified. Processing cannot continue because necessary information such as the C/I Size cannot be determined. Processing terminates.

User response:

Contact IBM Software Support. Provide the sysout listing for the JOB execution, including the Joblog. IBM Software Support may ask for additional listings from various utilities to identify the problem.

CKZ04645W First VOLSEQ Missing for Index Component DSN=comp.name

Explanation:

The first volume of a multi-volume set for the Index Component was not identified. Processing continues

without the Index Component to assist in reassembling spanned logical records. However, if any spanned records are encountered, deblocking errors are likely to occur.

User response:

No action is required.

CKZ04646E First Data Component VVR not a "Z" record, DSN=comp.name

Explanation:

The first volume's VVR for the Data Component was an unexpected type. Processing cannot continue because necessary information such as the C/I Size cannot be determined. Processing terminates.

User response:

Contact IBM Software Support. Provide the sysout listing for the JOB execution, including the Joblog. IBM Software Support may ask for additional listings from various utilities to identify the problem.

CKZ04646W First Index Component VVR not a "Z" record, DSN=comp.name

Explanation:

The first volume's VVR for the Index Component was an unexpected type. Processing continues without the Index Component to assist in reassembling spanned logical records. However, if any spanned records are encountered, deblocking errors are likely to occur.

User response:

No action is required.

CKZ04647E EXCP Error - cmd descr volser cc_hh_r dsn

Explanation:

An EXCP error occurred while reading the data set. Information is extracted from the standard IBM SYNAD Message. It is likely that the data set has been physically corrupted. Processing terminates.

User response:

Contact IBM Software Support. Provide the sysout listing for the JOB execution, including the Joblog. IBM Software Support may ask for additional listings from various utilities to identify the problem.

CKZ04647W EXCP Error - cmd descr volser cc_hh_r dsn

Explanation:

An EXCP error occurred while reading the data set. Information is extracted from the standard IBM SYNAD Message. It is likely that the data set has been physically corrupted. The nature of this error is not considered immediately critical. Processing continues.

User response:

No action is required. Informational warning message.

CKZ04649E **No paths available *volser* Reason:
*message_text***

Explanation:

While attempting to access the device, program CKZ00046 could not identify an available channel path to the device with *volser volser*.

User response:

Verify that channel paths are available to device by issuing MVS DISPLAY commands, such as **DM=DEV (xxxx)** and **DM=CHP (yy)**. If the device paths appear valid, contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04650I **Verification passed for
class_profile | Verification passed
for *class_profile* VOL(*volser*)**

Explanation:

The user ID that is running the COPY command has specific access to the profile *profile* and the class *class*.

User response:

No action is required.

CKZ04651I **No access to *class_profile* | No
access to *class_profile* VOL(*volser*)
| RACF No access to *class_profile* |
RACF No access to *class_profile*
VOL(*volser*)**

Explanation:

The user ID that is running the COPY command does not have specific access to the profile *profile* and the class *class*.

User response:

If message CKZ04622E was also issued, then grant the corresponding access to the user ID. Otherwise, no action is required.

CKZ04652I **SMS subcell information for
backed up catalog *list***

Explanation:

This informational message displays the BCS name, cluster name, component name, and sphere name from VVDS for a catalog that is being backed up.

User response:

No action is required.

CKZ04700I **PGM CKZ00047 PROGRAM REV=x**

Explanation:

Program CKZ00047 displaying its versioning information.

User response:

No action is required.

CKZ04701E **CKZ00047 Parameter Error.
parameter description**

Explanation:

CKZ00047 has detected invalid parameters from the calling routine. This is an internal error.

User response:

Contact IBM Software Support. Have the execution output listing available.

CKZ04704E **IARV64 REQUEST=GETSTOR Error
RC=*return_code* RSN=*reason_code*
requesting *number_of_segments*
1Mb segments LOC=*location***

Explanation:

An error occurred when a new 64-bit memory object is allocated. Processing terminates.

User response:

Contact IBM Software Support. Have the execution output listing available.

CKZ04706I **PGM CKZ00047 ????????
Processing Completed RC=xx
timestamp**

Explanation:

Program CKZ00047 processing is terminating with the above return-code.

User response:

If RC=00, None. If the Return-Code is any non-zero value, then contact IBM Software Support, and have the execution output listing available. There will be previous messages indicating the error causing the bad return code.

CKZ04710E **CKZ00047 ABENDED S-xxx |
CKZ00047 ABENDED U-xxxx**

Explanation:

Program CKZ00047 has suffered an abend and is taking appropriate recovery and cleanup actions. The requested function appearing in the CKZ04700I message has failed.

User response:

Contact IBM Software Support. Have the execution listings and the SYSUDUMP output available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04720E **I/O Error Reading Volume Label on
Device /xxxx**

Explanation:

Program CKZ00047 was unable to read the volume label at the indicated device address.

User response:

Determine if the device at the indicated address can be varied OFFLINE and ONLINE. The volume may be

uninitialized. If the volume can be successfully mounted, then contact IBM Software Support. Have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ04721E Validation on Device /xxxx failed.
Detected VOL=yyyyyy**

Explanation:

Program CKZ00047 read the volume label at the indicated device address and found a volume serial number that was different than what was expected.

User response:

Contact IBM Software Support. Have the execution output listing available.

**CKZ04740E SYSVTOC Exclusive Reserve
Required. | SYSZVVDS Exclusive
Reserve Required. | ** Volume
Restore Not Attempted ****

Explanation:

While preparing to RESTORE the VTOC and/or VVDS of the volume indicated by the CKZ04700I message, adequate RESERVE resources were not held. This is an internal error.

User response:

Contact IBM Software Support. Have the execution output listing available.

CKZ04748E Invalid Dump Record. ID ????????

Explanation:

While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00047 has determined that the logical contents of the sequential backup data set are invalid.

User response:

Contact IBM Software Support. Have the execution output listing available. If possible, save the backup data set indicated by this message in case it is requested by IBM Software Support.

**CKZ04749E Buffer Capacity Exceeded.
TYPE=????**

Explanation:

RESTORE processing has failed due to incorrect buffer size calculations. This is an internal error.

User response:

Contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ04750E ** EXCP I/O ERROR processing
the ?????????? ** | Track(CCHH)
Address: cchh | Synad: SYNAD
error text**

Explanation:

An uncorrectable error has occurred to the device being DUMPed or RESTORed while CKZ00047 was performing I/O using the EXCP access method.

User response:

Contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ04752E Error in structure of target volume.
| Description: text |
Location: location**

Explanation:

An error in the structure of a volume was found.

User response:

Contact IBM Software Support. Have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ04754E Device ??? - No Paths Available.
Reason: ?????? | Reason: IOS
Message text**

Explanation:

While attempting to access the device, CKZ00047 could not identify an available channel path to the device slated for DUMP or RESTORE processing.

User response:

Verify that channel paths are available to device by issuing MVS display commands such as D M=DEV(xxxx) and D M=CHP(yy). If device pathing appears valid, then contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ04755E Storage Subsystem for Device ???
Does not Support ECKD CCWs.**

Explanation:

CKZ00047 has detected an old technology DASD Control Unit that does not support hardware features that are minimally required by this. Such control units are typically used for supporting devices that pre-date 3380's. All control units for 3390's support ECKD transfer protocol.

User response:

No action is required. Db2 Cloning Tool cannot be used for this device.

CKZ04760I variable_text

Explanation:

This message displays informational statistics.

User response:

No action is required.

CKZ04761I *Informational messages*

Explanation:

Informational statistics regarding DUMP processing.

User response:

No action is required.

CKZ04761W **Note: Requested VVDS data set not in use.**

Explanation:

Informational warning regarding DUMP processing. The caller of CKZ00047 specified a non-standard data set name for the VVDS, which was not found on the volume. The correct data set name for the VVDS was found, and will be assumed as valid for DUMP processing.

User response:

No action is required. DUMP processing continues.

CKZ04761E **** ERROR ** Requested VVDS data set not found. | ** ERROR ** Required VVDS data set not found.**

Explanation:

A VVDS was not found on the volume that was being processed for dump. In addition, the volume was SMS managed, and/or contained VSAM data sets.

User response:

Verify that the volume is usable. If not, then a volume restore is in order. In either case, contact IBM Software Support, and have the execution output listings available.

CKZ04762I *variable_text*

Explanation:

This message displays informational statistics regarding RESTORE processing.

User response:

No action is required.

CKZ04763E **VTOC location has changed since the backup was taken | Volume dump for volser is unusable.**

Explanation:

This message displays informational statistics regarding RESTORE processing.

User response:

No action is required.

CKZ04779E **Internal error; LOC=location**

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ04780I **Data set record component is found in the VVDS**

Explanation:

The information about the catalog component is found in the VVDS.

User response:

No action is required.

CKZ04781W **Information for *component* not found in VTOC on the volume *volume* | Information for *component* not found in VVDS on the volume *volume***

Explanation:

The component *component* of the catalog residing on *volume* was not found when RENAME SAFE ran. Processing continues, but problems are possible for the RENAME RERUN command.

User response:

Contact IBM Software Support. Have available the listing containing this message and the information about the VTOC/VVDS for the corresponding volume.

CKZ04782W **Volume information cell is not found for dataset record | Volume information cell contains block size=0 for data set record**

Explanation:

The VVR for the corresponding catalog component is found, but information about the block size of the component is not defined. Processing continues, but problems are possible for the RENAME RERUN command.

User response:

Contact IBM Software Support. Have available the listing containing this message and the copy of the VVDS for the corresponding volume.

CKZ04783I **The information for data set record component: Block size: *size*. Max RBA: *rba*. Extended addressability: Y | N.**

Explanation:

The information for the catalog's component from the VVDS is displayed.

User response:

No action is required.

CKZ04807W **ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ04808E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found.
Processing terminates.

User response:

Check that the job's STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ04817E ERROR PROCESSING VVCNS:
message**

Explanation:

An error occurred during VVDS processing. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ04818E VOLSER: vvvvvv IS NOT ONLINE

Explanation:

The indicated volume is not online. Processing terminates.

User response:

Bring the volume online.

**CKZ04860E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn
LOC=lllll**

Explanation:

An error occurred during UCBLOOK processing. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ04861E IOSCAPU function ERROR;
RETURN CODE=nn REASON
CODE=nn LOC=lllll**

Explanation:

An attempt to use IOSCAPU to capture or uncapture a UCB failed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ05501I hh:mm:ss VOLUME CHECK
STARTED - PROGRAM REV=rrr |
hh:mm:ss VOLUME CHECK
COMPLETED; RETURN CODE=nnn**

Explanation:

COPYCHECK command processing message.

User response:

No action is required.

**CKZ05503I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ05504E DDNAME MISSING: ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

**CKZ05505E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ05505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ05507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ05508E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ05509E ERROR ACCESSING JOURNAL FILE; LOC=lllll
Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ05511E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL VOLUME PAIR RECORD(S) NOT FOUND
Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ05512E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLP RECORD IS WRONG VERSION
Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ05513E RECORD COUNT IS ZERO; LOC=lllll | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc LOC=lllll
Explanation:

There was a problem with the journal records needed to initiate the volume check. For the first format, the journal control record indicate no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ05515W THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY
Explanation:

The journal indicates that the COPY command did not complete successfully. Processing continues.

User response:

No action is required.

CKZ05531I COPY STEP WAS A SIMULATION
Explanation:

The journal indicates that the COPY command was a simulation. No volume checking will be done.

User response:

No action is required.

CKZ05540W COPIES DONE OUTSIDE OF DB2 CLONING TOOL; NO VOLUME CHECKING WILL BE DONE
Explanation:

The volume pairing was not initiated by the Db2 Cloning Tool COPY command.

User response:

No action is required.

CKZ05550E ERROR IN PARAMETERS FOR keyword
Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ05551E REQUIRED KEYWORD MISSING: keyword
Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ05553E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED
Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ05554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ05556E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ05558E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ05586I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

**CKZ05601I hh:mm:ss VOLUME STATUS
STARTED - PROGRAM REV=rrr |
hh:mm:ss VOLUME STATUS
COMPLETED; RETURN CODE=nnn**

Explanation:

Volume status processing message.

User response:

No action is required.

**CKZ05607W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ05608E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ05630I VOLUME PAIRS STATUS

Explanation:

The status of each source/target volume serial pair follows this message.

User response:

No action is required.

**CKZ05638I UNEXPECTED RESULTS FROM
FCQUERY FQMAP; VOLSER= volser**

Explanation:

An ANTRQST FCQUERY FQMAP call returned unexpected results.

User response:

Please report this message to IBM Software Support.

CKZ05641W TIME LIMIT EXCEEDED

Explanation:

The WAIT time for the command has been reached. Some volume pairs may still be in a copy relationship.

User response:

No action is required. If desired, rerun the COPYCHECK command with a higher WAIT time specified.

**CKZ05643I ANTRQST LEVEL=nn; ESSRVCS
LEVEL=nnn**

Explanation:

The level returned by ANTRQST REQUEST=LEVEL

User response:

No action is required.

**CKZ05644E ANTRQST LEVEL NOT SUPPORTED;
LEVEL=nn**

Explanation:

For FlashCopy support, the level must be greater than four (4). For SnapShot support, the level must be greater than one (1). For PPRC support, the level must be one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response:

Check with your system programmer for upgrading the system.

CKZ05645E system task NOT ACTIVE

Explanation:

An ANTRQST request failed because a 'system task' is not active. Processing terminates.

User response:

Start the indicated system task.

CKZ05646E ANTRQST DID NOT RETURN ANY INFORMATION; RESTART system task SYSTEM TASK

Explanation:

The ANTRQST REQUEST=FCQUERY did not receive information for a device, or, REQUEST=PQUERY did not receive information for a device. Processing terminates.

User response:

Restart the indicated system task.

CKZ05647E ANTRQST ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=lllll

Explanation:

An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ05648E type VOLUME volser ATTRIBUTES HAVE CHANGED;

Explanation:

The indicated volume has physically moved since COPY. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the moved volume. Processing terminates.

User response:

Do not use COPYCHECK after SOURCE or TARGET volumes have been moved from where they were at the time of COPY.

CKZ05649E type VOLUME volser IS NOT ONLINE

Explanation:

The indicated volume is not online. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the volume. Processing terminates.

User response:

Do not use COPYCHECK if SOURCE or TARGET volumes are offline.

CKZ05650E UNRECOGNIZED STATUS FROM PQUERY

Explanation:

The device status returned by the indicated query was not recognized. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ05651I VOLSER PAIR source_volser/target_volser COPY STILL IN PROGRESS, numeric% COMPLETED | VOLSER PAIR source_volser/target_volser COPY STILL IN PROGRESS, FCNOCOPY | VOLSER PAIR source_volser/target_volser COPY STILL IN PROGRESS, IN SYNC | VOLSER PAIR source_volser/target_volser COPY STILL IN PROGRESS, NOT AVAILABLE

Explanation:

FCQUERY indicates that the volumes are still in a copy relationship. If the background copy was initiated, *numeric* indicates the percentage of the copy that is completed. If FCNOCOPY was used, no background copy was initiated and the second form of this message is issued. If the source and target are in the same status, and PPRC copy is in synchronization, the third form of this message is issued. If FlashCopy is still in progress, and the percentage of the copy progress is not available, the fourth form of the message is issued.

User response:

No action is required.

CKZ05652I VOLSER PAIR source_volser/target_volser FCWITHDRAW ISSUED | VOLSER PAIR source_volser/target_volser PDELPAIR ISSUED

Explanation:

FCWITHDRAW or PDELPAIR has been issued for the volumes.

User response:

No action is required.

CKZ05653I VOLSER pair: source_volser/target_volser FCWITHDRAW indicated the relationship no longer existed

Explanation:

The FCWITHDRAW for the volumes indicated that there was no FlashCopy relationship between the two volumes. There is a small window between the check for FlashCopy relationships on the volumes and when the FlashCopy WITHDRAW is done. If the relationship ends between the check and FlashCopy WITHDRAW, this message is displayed to document the event.

User response:

No action is required.

CKZ05660E UCBLOOK ERROR; RETURN CODE=nn REASON CODE=nn

Explanation:

An error occurred using the UCBLook macro.
Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ05701I hh:mm:ss EMC VOLUME STATUS
STARTED - PROGRAM REV=rrr |
hh:mm:ss EMC VOLUME STATUS
COMPLETED; RETURN CODE=nnn**

Explanation:

EMC volume status processing message.

User response:

No action is required.

**CKZ05707W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record.
Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ05708E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found.
Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If
unable to resolve the problem, contact IBM Software
Support.

**CKZ05730I VOLUME PAIRS STATUS source
volser/target volser status**

Explanation:

The status of each source/target volume serial pair is
displayed.

User response:

No action is required.

**CKZ05738E VOLP ENTRY NOT FOUND FOR
VOLSER=volser**

Explanation:

An internal processing error has occurred. Processing
terminates.

User response:

Contact IBM Software Support. Have available the
listing that contains this message.

**CKZ05739I EMCSNAPI COMPLETED; RETURN
CODE=nnn**

Explanation:

A call to the EMCSNAPI module has completed with a
return code of nnn.

User response:

No action is required.

CKZ05740E INTERNAL ERROR; reason

Explanation:

An internal error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the
listing that contains this message.

CKZ05741W TIME LIMIT EXCEEDED

Explanation:

The WAIT time for the command has been reached.
Some volume pairs may still be in a snap relationship.

User response:

No action is required. If desired, rerun the
COPYCHECK command with a higher WAIT time
specified.

**CKZ05742W STOP SNAP PENDING; SNAPS ARE
STILL ACTIVE**

Explanation:

A STOPSNAP was requested and 10 minutes after the
STOP SNAP commands have been issued there are still
some volume pairs in a snap relationship.

User response:

Determine why the snap relationships have not
stopped.

CKZ05743I module-name VERSION version

Explanation:

This message reports the version information retrieved
from the indicated module.

User response:

No action is required.

CKZ05745E EMCSCF NOT ACTIVE; LOC=llll

Explanation:

An EMC request failed because the system task,
EMCSCF, is not active. Processing terminates.

User response:

Start the system task, EMCSCF.

**CKZ05746E module-name ERROR; request
type R15=nnnn X'hhhh' LOC=llll
VOLSER=volser**

Explanation:

The call to the indicated module for the indicated
request type failed. The return code is displayed in
decimal and hexadecimal format. 'lllll' is an internal
indicator of where the problem occurred. Processing
terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ05747E **module-name ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=lllll VOLSER=volser**

Explanation:

The call to the indicated module for the indicated request type failed. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ05748E **type VOLUME volser ATTRIBUTES HAVE CHANGED; attributes**

Explanation:

The indicated volume has physically moved since COPY. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the moved volume. Processing terminates.

User response:

Do not use COPYCHECK after SOURCE or TARGET volumes have been moved from where they were at the time of COPY.

CKZ05749E **type VOLUME volser IS NOT ONLINE attributes**

Explanation:

The indicated volume is not online. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the volume. Processing terminates.

User response:

Do not use COPYCHECK if SOURCE or TARGET volumes are offline.

CKZ05751I **VOLSER PAIR source volser/target volser SNAP STILL IN PROGRESS, nnn% COMPLETED - type**

Explanation:

The volumes are still in a snap relationship. 'nnn%' indicates the percent of the copy that is completed. 'type' indicates if this is a DIFFERENTIAL snap or a snap with no background copy.

User response:

No action is required.

CKZ05752I **VOLSER PAIR source volser/target volser STOP SNAP ISSUED**

Explanation:

STOP SNAP has been issued for the volumes.

User response:

No action is required.

CKZ05756I **CURRENT SNAP RELATIONSHIPS FOR type VOLSER: volser volser relationship information**

Explanation:

The current snap relationships for the volser are listed. 'type' indicates if the volser is a SOURCE or TARGET volume.

User response:

No action is required.

CKZ05760E **UCBLOOK ERROR; RETURN CODE=nn REASON CODE=nn**

Explanation:

An error occurred using the UCBLOOK macro. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ06001I **hh:mm:ss BACKINFO REFORMAT STARTED - PROGRAM REV=rrr | hh:mm:ss BACKINFO REFORMAT COMPLETED; RETURN CODE=nnn**

Explanation:

BACKINFO-REFORMAT processing message.

User response:

No action is required.

CKZ06004E **DDNAME MISSING: ddname | OPEN FAILED FOR DDNAME: ddname**

Explanation:

ddname was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ06006E **ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the PARMLIB member that controls execution of Db2 Cloning Tool.

CKZ06007W **ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ06008E UNABLE TO LOAD PROGRAM:
 program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ06019E THE keyword DOES NOT HAVE A
 LRECL OF 80, DDNAME: ddname**

Explanation:

The data set allocated to the ddname does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response:

Change the data set allocated to the ddname to have a LRECL of 80.

**CKZ06020E UNKNOWN RECORD TYPE FOUND
 IN BACKINFO record**

Explanation:

An unknown record type was found in the backinfo data set. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06021E DUPLICATE type VOLSER FOUND
 IN BACKINFO; VOLSER: vvvvvv**

Explanation:

The indicated volsr was found multiple times in the backinfo data set. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06022E DUPLICATE CATALOG DSN FOUND
 IN BACKINFO; DSN: dataset**

Explanation:

The indicated catalog DSN was found multiple times in the backinfo data set. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06023E NO VOLMAP RECORDS FOUND IN
 BACKINFO**

Explanation:

No VOLMAP type records were found in the backinfo data set. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06024E INVALID type DEVN FOUND IN
 BACKINFO; DEVN: nnnn reason**

Explanation:

An invalid device number was found in the backinfo data set. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06025E BACKUP VOLSER: vvvvvv IS NOT
 ONLINE**

Explanation:

The indicated backup volume volsr is not online. Processing terminates.

User response:

Bring the indicated volume online or use CLIP-IF-OFFLINE(Y) if the volume is offline and needs to be clipped.

**CKZ06026E BACKUP VOLSER: vvvvvv IS NOT
 ONLINE BUT ITS SPECIFIED
 DEVN: dddd IS ONLINE WITH
 VOLSER: vvvvvv**

Explanation:

The indicated backup volume volsr is not online, but its specified device number is online with a different volsr. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06027E SPECIFIED BACKUP VOLSER:
 vvvvvv IS ONLINE ON DEVN: dddd
 WHICH IS NOT ITS SPECIFIED
 DEVN: dddd**

Explanation:

The indicated backup volume volsr is online, but it is on a different device number than specified in the backinfo data set. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06028E UNABLE TO GENERATE BACKUP
VOLSER FOR SOURCE VOLSER:
vvvvvv; NO MATCHING VOLSER
RENAME MASK FOUND**

Explanation:

A backup volume volser needs to be generated, but there is no entry in the VOLSER-RENAME-MASKS keyword that matches the volser of its corresponding source volume. Processing terminates.

User response:

Add a entry to the VOLSER-RENAME-MASKS keyword that matches the indicated source volser.

**CKZ06029E SPECIFIED BACKUP DEVN: dddd
IS NOT DEFINED TO Z/OS**

Explanation:

The device number specified for a backup volume in the backinfo data set is not defined to z/OS. Processing terminates.

User response:

Check that the device number specified is correct and the job is running on a z/OS system where the device is defined.

**CKZ06030E BACKUP VOLSER NOT SPECIFIED
FOR SOURCE VOLSER: vvvvvv**

Explanation:

A unique volser was not specified for a backup volume in the backinfo data set and the CLIP-IF-OFFLINE(Y) keyword was not used. Processing terminates.

User response:

Determine if the CLIP-IF-OFFLINE(Y) keyword should be used or correct the backinfo data set to have a unique volser for the backup volume.

**CKZ06031E DUPLICATE BACKUP VOLSER:
vvvvvv FOUND FOR SOURCE
VOLSER: vvvvvv AND SOURCE
VOLSER: vvvvvv**

Explanation:

The backinfo data set has two VOLMAP records for different source volumes that have the same backup volser specified. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06032E SOURCE VOLSER: vvvvvv IS ALSO
USED AS A BACKUP VOLSER FOR
SOURCE VOLSER: vvvvvv**

Explanation:

The backinfo data set has a source volser that is also used as a backup volser for a different source volume. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06033E VOLSER: vvvvvv FOR UCAT:
usercatalog DOES NOT MATCH
ANY SOURCE VOLSER**

Explanation:

The backinfo data set has a UCAT record for the indicated user catalog, but the volser specified for it is not a source volser in a VOLMAP record. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06034E USERCATALOGS ENTRY:
usercatalog DOES NOT MATCH
ANY UCAT ENTRY FOUND IN
BACKINFO**

Explanation:

The indicated user catalog was specified in the USERCATALOGS keyword, but the backinfo data set does not have a UCAT entry for that user catalog. Processing terminates.

User response:

Check that a correct backinfo data set is being used and the user catalog has been specified correctly in the USERCATALOGS keyword.

**CKZ06035I NO BACKUP VOLUMES NEED TO BE
CLIPPED**

Explanation:

All the backup volumes are already online with their expected volsers.

User response:

No action is required.

**CKZ06036E NO BACKUP VOLSER SPECIFIED
IN BACKINFO FOR SOURCE
VOLSER: vvvvvv BUT ITS
SPECIFIED DEVN: dddd IS
ONLINE WITH VOLSER: vvvvvv**

Explanation:

The VOLMAP record in the backinfo data set for source volser has no backup volume volser. This condition means that the backup volume is expected to be offline but the backup volume device was found to be online with the indicated volser. Processing terminates.

User response:

Check that a correct backinfo data set is being used. If this is a rerun of a prior failed CLIP-IF-OFFLINE(Y) run, specify the RESUME keyword.

**CKZ06037I BACKUP VOLUME ON DEVN: dddd
APPEARS TO HAVE ALREADY
BEEN CLIPPED TO VOLSER: vvvvvv**

Explanation:

The CLIP-IF-OFFLINE(Y) and RESUME keywords have been specified and the indicated backup volume appears to have already been clipped to the indicated volser.

User response:

No action is required.

**CKZ06038I BACKUP VOLUME ON DEVN: dddd
WILL BE CLIPPED TO {GENERATED
| SPECIFIED} VOLSER: vvvvvv**

Explanation:

The backup volume on the indicated device will be clipped to the indicated generated or specified volser.

User response:

No action is required.

**CKZ06039I BACKUP VOLUME ON DEVN: dddd
IS ALREADY ONLINE WITH
SPECIFIED VOLSER: vvvvvv**

Explanation:

The backup volume on the indicated device is already with the specified volser.

User response:

No action is required.

**CKZ06040E CKZ00900 UNEXPECTED
RESULTS; error text**

Explanation:

An unexpected condition occurred calling program CKZ00900. Error text has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06041E INTERNAL ERROR; LOC=lllll

Explanation:

An internal error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ06042I VOLUME PAIRS BEING USED: list
of pairs**

Explanation:

The listed pairs were found in the backinfo data set.

User response:

No action is required.

**CKZ06043E There is a blank storage group
name for a volume in the backinfo
file.**

Explanation:

A blank storage group name was found for a volume in the backinfo data set. The USERSGDEFS-DDN keyword has been specified, and the backinfo data set must have storage group names for all the volumes. Processing terminates.

User response:

Check that a correct backinfo data set is being used and that the correct version of the program that creates the backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06050E ERROR IN PARAMETERS FOR
keyword**

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ06051E REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ06052E THE SAME DDNAME HAS BEEN
SPECIFIED FOR MULTIPLE
KEYWORDS: ddname**

Explanation:

The indicated ddname has been specified in multiple keywords. The specified ddnames must all be different. Processing terminates.

User response:

Specify different ddnames in the keywords.

**CKZ06053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ06054E **KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ06056E **NOTHING SPECIFIED FOR KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ06057E **DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry**

Explanation:

The indicated entry for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ06058E **INVALID VALUE IN KEYWORD: keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ06060E **UCBLOOK ERROR; RETURN CODE=nn REASON CODE=nn LOC=lllll**

Explanation:

An error occurred during UCBLOOK processing. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ06064E **INVALID VOLSER: volser IN KEYWORD: keyword**

Explanation:

The volume serial number specified is invalid. Processing terminates.

User response:

Correct the volser specification.

CKZ06068E **UNMATCHED ENTRIES IN KEYWORD: keyword**

Explanation:

For USERCATALOGS, there must be a source BCS followed by a target BCS. An uneven number of BCSs was specified. For VOLSER-RENAME-MASKS, there must be a source mask followed by a backup mask. An uneven number of masks was specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ06081I **DSNS FOR KEYWORD: keyword list of dsns**

Explanation:

Parsing found the listed dsns for the keyword.

User response:

No action is required.

CKZ06082I **VOLSER RENAME MASK PAIRS FOR KEYWORD: keyword list of mask pairs**

Explanation:

Parsing found the listed mask pairs for the keyword.

User response:

No action is required.

CKZ06083W **VOLSER-RENAME-MASKS SPECIFIED BUT WILL NOT BE USED DUE TO CLIP-IF-OFFLINE(N)**

Explanation:

The VOLSER-RENAME-MASKS keyword has been specified, but CLIP-IF-OFFLINE(N) has also been specified or defaulted to. The volser rename masks will not be used.

User response:

None, or remove the VOLSER-RENAME-MASKS keyword, or use CLIP-IF-OFFLINE(Y).

CKZ06086I **VALIDATING KEYWORD: keyword**

Explanation:

Parsing is checking the indicated keyword in the command.

User response:

No action is required.

CKZ06501I **COPY-BY-DS started - program rev=revision**

Explanation:

The COPY-BY-DS command started with the program level that is listed in the message.

User response:

No action is required.

CKZ06502I **COPY-BY-DS completed - return code = return_code**

Explanation:

The COPY-BY-DS command completed with the return code that is listed in the message.

User response:

No action is required.

CKZ06503E Allocation failure for DDname: *ddname*

Explanation:

Dynamic allocation failed for the ddname that is listed in the message.

User response:

Contact IBM Software Support.

CKZ06504E DDname missing: *ddname*

Explanation:

A keyword indicated that the given ddname was supplied, but the ddname was not found.

User response:

Contact IBM Software Support.

CKZ06505E Open failed for DDname: *ddname*

Explanation:

An open failed for the ddname that is listed in the message.

User response:

Ensure that the ddname is correct and that the file is accessible via ISPF or other means. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06506E No data sets found for given RENAME-MASKS values

Explanation:

The given RENAME-MASKS values returned no data sets.

User response:

Check the values that are provided in the keyword RENAME-MASKS. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06507W Error calling *routine_name*; function=*function* R15=*return_code*

Explanation:

A call to an internal routine to translate storage into hexadecimal for printing failed with the given return code and function that are listed in the message. Processing continues.

User response:

This message is a warning and can be ignored unless the hexadecimal values are essential for problem resolution. If you need assistance, contact IBM Software Support.

CKZ06509E Error in RENAME-MASKS: *mask*

Explanation:

The supplied rename mask contains errors. Processing stops.

User response:

Check the values in the RENAME-MASKS keyword. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06510E ATTACHX failed RC *return_code* RSN *reason_code*

Explanation:

An attempt to ATTACH the COPY-BY-DS driver failed. The return code and reason from the ATTACHX macro are supplied in the message.

User response:

Contact IBM Software Support.

CKZ06511E Unable to establish ESTAEX; r15=*return_code*

Explanation:

An attempt to set up a recovery environment via the ESTAEX macro failed with the return code that is listed in the message.

User response:

Contact IBM Software Support.

CKZ06512I Error recovery in process

Explanation:

The recovery routine was invoked after an error.

User response:

No action is required.

CKZ06513I Total number of data sets to be copied: *number_of_data_sets*

Explanation:

This message provides the total number of data sets to be copied.

User response:

No action is required.

CKZ06515I *Informational_message_detail_header*

Explanation:

This message provides header information for a report. Information is included for the COPY command task number, COPY command number, type of data set, the source data set name, the source volume, the target data set name, and the rename mask.

User response:

No action is required.

CKZ06516I Source data sets to be copied, on volumes | Target data sets to be created | Target storage groups

Explanation:

This message provides a header line for a data set list report, detailing data sets to be copied, source volumes, and target data sets to be created, plus target storage groups (if applicable).

User response:

No action is required.

CKZ06517E **Empty or dummy ddname: *ddname***

Explanation:

The file represented by the ddname that is listed in the message is empty and cannot be processed. Processing stops.

User response:

Specify a valid file and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06518E **ERROR: crash completion_code at program_status_word_address routine_name routine_offset - ***** TERMINATING *******

Explanation:

An error occurred that was captured by a recovery routine. The system completion code and program status word address are provided in the message. If available, the routine name and offset are also provided. Processing stops after recovery.

User response:

Contact IBM Software Support.

CKZ06519E **RENAME-MASKS-DDN lrecl not = 80, DDname: *ddname***

Explanation:

Logical record length must be 80 for the ddname that is provided in the message.

User response:

Correct the ddname. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06520I **Error recovery complete**

Explanation:

Recovery processing is complete. Processing stops.

User response:

No action is required.

CKZ06521E **COPY-BY-DS incomplete due to previous errors**

Explanation:

Copy by data set processing did not successfully complete.

User response:

Review and resolve other error messages in the message file. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06522E **Volume lookup for target data sets found *target_number* files; does not match source data sets found *source_number***

Explanation:

The number of source data sets (less migrated and excluded data sets) does not match the number of target data sets.

User response:

Contact IBM Software Support.

CKZ06526E **ERROR CALLING CKZ01VV2 TABLE=*tablename* FUNC=*function* R15=*register15* R0=*register0* LOC=*location***

Explanation:

An error occurred when processing an internal table. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06529I **Migrated data sets (*number_of_migrated_data_sets*) will not be copied due to MIGRATED-DSN(SKIP)**

Explanation:

Migrated data sets will not be copied because MIGRATED-DSN(SKIP) was specified. The number of migrated data sets remaining after EXCLUDE processing is provided in the message.

User response:

No action is required.

CKZ06530E **Migrated data sets (*number_of_migrated_data_sets*) prevent further processing**

Explanation:

Migrated data sets cannot be copied. The number of migrated data sets remaining after EXCLUDE processing is provided in the message. A list of the migrated data sets that are not excluded will be displayed.

User response:

Either recall the migrated data sets or exclude them from processing. The migrated data sets can be excluded by using the EXCLUDE-SRCNAME-MASKS keyword or the MIGRATED-DSN(SKIP) keyword.

CKZ06531E ***** ERROR *** - see following ADRDSSU message - *** ERROR *****

Explanation:

An error message from ADRDSSU follows this message. Normally, this message results in a return code of 8 for the COPY-BY-DS command.

User response:

Use the ADRDSSU message for problem determination and correct the problem before resubmitting the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06532I **Excluded data sets (*number*) will not be copied**

Explanation:

The number of data sets that are listed in the message will not be copied because they match an EXCLUDE data set name mask.

User response:

No action is required.

CKZ06536I **Source volumes written to the journal file:**

Explanation:

A listing of the source volumes that were found and written to the journal follows this message. The detail lines are provided in message CKZ06538I. If this run is in SIMULATE mode, it is indicated in the message.

User response:

No action is required.

CKZ06537I **Target volumes written to the journal file:**

Explanation:

A listing of the target volumes that were found and written to the journal follows this message. The detail lines are provided in message CKZ06538I. If this run is in SIMULATE mode, it is indicated in the message.

User response:

No action is required.

CKZ06538I ***volume***

Explanation:

This message is displayed with messages CKZ06536I and CKZ06537I and provides a list of volumes, one per line.

User response:

No action is required.

CKZ06540E **Duplicate journal file entry; *loc=location***

Explanation:

A duplicate journal record was found at the internal location that is listed in the message. It is possible that multiple mask pair entries in the RENAME-MASKS keyword are resolving to the same target data set name.

User response:

Modify the RENAME-MASKS keyword to remove the mask pair entry that caused the duplicate target data set name. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06541E **Error accessing journal file; *loc=location***

Explanation:

An error occurred when accessing the journal file at the internal location that is listed in the message.

User response:

Contact IBM Software Support.

CKZ06542E **Error CKZ0900 : *function additional_information***

Explanation:

A parsing error occurred in the function that is listed in the message. If available, additional text is provided in the message.

User response:

Contact IBM Software Support.

CKZ06543E **Update of journal file record failed**

Explanation:

An attempt to update a journal record failed.

User response:

Contact IBM Software Support.

CKZ06546E **INTERNAL ERROR; *error_text loc=location***

Explanation:

This is an internal error. *error_text* contains a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06548I **Options in effect for COPY-BY-DS job step:**

Explanation:

A listing of the options in effect follows.

User response:

No action is required.

CKZ06550E **Error in parameters for keyword: *keyword additional_information***

Explanation:

An error occurred for the keyword that is listed in the message. If available, additional text is provided in the message.

User response:

Check the documentation for the keyword syntax and correct it before resubmitting the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06551E **One of: *keyword_1* or *keyword_2* is required****Explanation:**

One of the two listed keywords is required.

User response:

Provide one of the required keywords and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06552E **DSName mask invalid: *mask_name*****Explanation:**

The data set name mask that is listed in the message is invalid.

User response:

Correct the data set name mask and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06553E **Keyword: *keyword* maximum length: *max_length* exceeded****Explanation:**

The keyword that is listed in the message exceeds the allowed maximum length.

User response:

Correct the keyword and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06554E **ONLINE keyword not supported in this release****Explanation:**

Only OFFLINE clones are supported by COPY-BY-DS in this release.

User response:

Correct the keyword and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06555E **DSName mask has invalid characters: *offset data_set_name_mask*****Explanation:**

The data set name mask that is listed in the message has an invalid character at the offset that is provided in the message.

User response:

Correct the mask and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06556E **Unable to process message *message* above****Explanation:**

An internal error occurred with the message table for the given message.

User response:

Contact IBM Software Support.

CKZ06557E **DSName mask too long: *length data_set_name_mask*****Explanation:**

The data set name mask that is listed in the message is too long.

User response:

Correct the mask and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06558E **Invalid value *value* in keyword: *keyword*****Explanation:**

The value that is listed in the message is invalid for the given keyword.

User response:

Correct the value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06559E **Keyword value has invalid characters: *value*****Explanation:**

The keyword value that is listed in the message contains invalid characters.

User response:

Correct the value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06560E **Keyword value too long: *value*****Explanation:**

The keyword value that is listed in the message is too long.

User response:

Correct the value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06561E **Keyword value cannot be zero: *value*****Explanation:**

The keyword value that is listed in the message cannot be zero.

User response:

Correct the value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06562E Keyword *keyword* value must be 1-3 numeric digits

Explanation:

The value for the keyword that is listed in the message must be 1-3 numeric digits.

User response:

Correct the keyword value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06563E Keyword *keyword* value must be REQ, PREF, or NONE

Explanation:

The keyword that is listed in the message must be REQ, PREF, or NONE.

User response:

Correct the keyword value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06564E Keyword *keyword* value cannot be greater than *value*

Explanation:

The keyword that is listed in the message cannot have a value greater than *value*.

User response:

Correct the keyword value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06565I DSS-COPY-COMMANDS * DSNS-PER-COPY = *number* - may cause storage and other resource-related errors during program execution

Explanation:

The supplied values for DSS-COPY-COMMANDS and DSNS-PER-COPY multiplied together give a product of *number*. This might cause storage and other resource-related errors during execution of the program that is listed in the message.

User response:

No action is required.

CKZ06566E Keyword PGM value must be *value*

Explanation:

The value that is listed in the message for the PGM subkeyword must be ADRDSSU.

User response:

Correct the value and resubmit the job. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06567I Recommend DSS-COPY-COMMANDS * DSNS-PER-COPY = 1400 or less

Explanation:

The recommended maximum for DSS-COPY-COMMANDS x DSNS-PER-COPY is 1400.

User response:

No action is required.

CKZ06568E NOTHING SPECIFIED FOR KEYWORD: *keyword*

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ06569E KEYWORD: *keyword* MAXIMUM LENGTH: *nnn* EXCEEDED

Explanation:

The operand that was entered for a keyword exceeded the maximum length allowed for the operand. *nnn* is the maximum allowed length for the keyword. Processing terminates..

User response:

Correct the length of the keyword's operand.

CKZ06570E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: *keyword*

Explanation:

Multiple operands were detected for a keyword, but only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ06571E CKZ01SMF ERROR; RETURN CODE= *return_code* LOC: *location* INVALID *sms_type* CLAS NAME: *class_entry*

Explanation:

An error occurred using CKZ01SMF to verify the SMS class that was specified. *location* is the internal location where the error occurred. *sms_type* indicates MGMT or STOR. Processing terminates.

User response:

Correct the indicated SMS class entry.

CKZ06572E FCTOPPRCPRIARY KEYWORD IS NOT COMPATIBLE WITH FASTREP(NONE)

Explanation:

The FASTREP parameter must be set to PREF or REQ to use FCTOPPRCPRIARY.

User response:

Correct the input and resubmit the job.

CKZ06573E **REQUIRED INI SECTION/TOKEN MISSING: SECTION=*section* TOKEN=*token* | REQUIRED INI VALUE MISSING FOR SECTION=*section* TOKEN=*token* | INVALID INI VALUE FOR SECTION=*section* TOKEN=*token***

Explanation:

An error occurred when validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

CKZ06574E **FCTOXRCPPRIMARY KEYWORD IS NOT COMPATIBLE WITH FASTREP(NONE)**

Explanation:

The FASTREP parameter must be set to PREF or REQ to use FCTOXRCPPRIMARY.

User response:

Correct the input and resubmit the job.

CKZ06606E **ERROR CALLING CKZ01VV2 TABLE=*tablename* FUNC=*function* R15=*register15* R0=*register0* LOC=*location***

Explanation:

An error occurred when processing an internal table. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06611W **Warning returned by CSI: Module: *module* Reason: *reason* Return code: *return_code* Reg 15: *value_in_R15***

Explanation:

A call to the Catalog Search Interface (CSI) returned a warning with the given information.

User response:

Check the data set causing the error. If unable to determine the cause of this error, contact IBM Software Support.

CKZ06612E **Error returned by CSI: Module: *module* Reason: *reason* Return code: *return_code* Reg 15: *value_in_R15***

Explanation:

A call to the Catalog Search Interface (CSI) returned an error with the given information.

User response:

Contact IBM Software Support.

CKZ06613E **Internal error with CSI record: *error_text***

Explanation:

This is an internal error that occurred with a CSI record. *error_text* contains a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06615E **Internal error error - Reason: *reason_code* - contact Support**

Explanation:

An internal error occurred with input parameters for the CSI driver program. The reason code is listed in the message. Processing stops.

User response:

Contact IBM Software Support.

CKZ06618E **Error indication is set for entry with name: *data_set_name* Reason code: *reason_code* Return code: *return_code* CSI Entry: *csi_entry***

Explanation:

The CSI entry with the data set name that is listed in the message has encountered an error.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06620E **Target data set *data_set_name* was not found after copy**

Explanation:

The target data set was not found in the CSI after the copy process.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06621E **Cluster component is not found for *data_set_name* LOC: *location* | Data component is not found for *data_set_name* LOC: *location***

Explanation:

The cluster or data component was not found in the CSI for the data set that is listed in the message.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ06622E **Maximum amount of volumes is exceeded for *data_set_name* LOC: *location***

Explanation:

The number of volumes for the data set that is listed in the message exceeds 200.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ06623E More than one CSI record for
data_set_name LOC: location**

Explanation:

More than one CSI record was found for the data set name that is listed in the message.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ06701I DSSU COPY-BY-DS driver started -
CKZ00067 rev= revision [**
SIMULATION **]**

Explanation:

Execution started for the COPY-BY-DS driver program. The program revision number is listed in the message. If SIMULATE mode was specified, it is noted in the message.

User response:

No action is required.

**CKZ06702I DSSU COPY-BY-DS driver
completed**

Explanation:

Execution is complete for the COPY-BY-DS driver program.

User response:

No action is required.

**CKZ06703I ** CKZ00067 Task ID id_number
started [** SIMULATION **]**

Explanation:

A copy task with the ID number that is listed in the message started, with a set of ADRDSSU COPY commands. If SIMULATE mode was specified, it is noted in the message.

User response:

No action is required.

**CKZ06704I ** CKZ00067 Task ID id_number
completed - return code =
return_code**

Explanation:

A copy task has completed a set of ADRDSSU COPY commands with the return code that is listed in the message.

User response:

No action is required.

**CKZ06712E Unable to establish ESTAEX; r15=
return_code**

Explanation:

An ESTAEX macro failed with the give return code. Error recovery could not be established in the COPY-BY-DS driver. Processing stops.

User response:

Contact IBM Software Support.

**CKZ06715I Skipping ADRDSSU with files
ddname_1 and ddname_2 due to
** SIMULATION **'**

Explanation:

SIMULATE mode was specified; therefore, the actual ADRDSSU COPY call is skipped.

User response:

No action is required.

**CKZ06716I Calling ADRDSSU with
TYPRUN=NORUN due to **
SIMULATION ****

Explanation:

With SIMULATE mode, and if a DEBUG067 DD card is present, ADRDSSU is called with TYPRUN=NORUN for a more complete simulation.

User response:

No action is required.

**CKZ06721E ADRDSSU copy failed; return code:
return_code**

Explanation:

The ADRDSSU COPY failed with the return code that is listed in the message.

User response:

Check for other error messages to determine the exact problem. If unable to determine the cause of this error, contact IBM Software Support.

**CKZ07000I RESTORE-FROM-DUMPTAPES
started - program rev=revision**

Explanation:

This message indicates the beginning of processing for RESTORE-FROM-DUMPTAPES, and includes the program revision level.

User response:

No action is required.

**CKZ07002I RESTORE-FROM-DUMPTAPES
completed; return
code=return_code**

Explanation:

This message indicates the end of processing for RESTORE-FROM-DUMPTAPES and includes the return code.

User response:
No action is required.

CKZ07004E **ddname missing: *ddname***

Explanation:
An open failed for the ddname that is listed in the message, or the ddname is missing.

User response:
If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07005E **Open failed for ddname: *ddname***

Explanation:
An open failed for the ddname that is listed in the message.

User response:
If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07008E **Unable to load program:
*load_module***

Explanation:
The load macro failed for the load module that is listed in the message.

User response:
Ensure that the STEPLIB contains the load module, or that the load module is available via normal system search. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07010W **Error calling CKZ01HEX;
function=function
*R15=return_code***

Explanation:
An error occurred when attempting to format output into hexadecimal for display. The error occurred in function *function* and with return code *return_code*.

User response:
If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07011E **Error accessing STATUS file;
*loc=location***

Explanation:
A STATUS file open failed at the given location (for internal use only).

User response:
Contact IBM Software Support.

CKZ07012E **Duplicate STATUS file entry;
*loc=location***

Explanation:
An attempt to insert the initial control record failed because a duplicate record was found.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07013E **STATUS file control record not found. *message_text***

Explanation:
If the RERUN keyword was specified, the control record must be present during initialization. If the RERUN keyword was not specified, an attempt to update the control record failed.

User response:
Contact IBM Software Support.

CKZ07014E **STATUS file control record version
record_version is not current with
program version *program_version***

Explanation:
An older STATUS file format was specified with a newer version of RESTORE-FROM-DUMPTAPES.

User response:
If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07015E **STATUS file control record
mismatch - *type=type* prevents
use of RERUN**

Explanation:
The control record in the STATUS file does not match the current RESTORE-FROM-DUMPTAPES command input.

User response:
If RERUN is required, it is possible that the wrong STATUS file is being used to RERUN the job; locate and specify the correct STATUS file. If RERUN is not required, remove the RERUN keyword, delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07016E **STATUS file control record shows
Status-Complete - prevents use of
RERUN**

Explanation:
The RERUN keyword was used, but the control record in the STATUS file shows that the clone finished with RC=0 or 4. Therefore, the clone cannot be rerun.

User response:
Remove the RERUN keyword, delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07017E **STATUS file restore records not found - prevents use of RERUN**

Explanation:

There is no rerun information other than the control record.

User response:

Remove the RERUN keyword, delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07018E **Keyword *keyword* not specified and default *ddname* *ddname* not supplied**

Explanation:

If the default ddnames for BACKINFO, VOLPAIRS, and UCATS are not supplied in the JCL, then for each of those not supplied, the appropriate BACKINFO-DDN, VOLPAIRS-DDN, or USERCATALOGS-DDN keyword must be specified.

User response:

Use the default ddnames, or supply the correct DDN keyword. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07019E ***ddname* *ddname* is not LRECL=80**

Explanation:

The backinfo data set (or the BACKINFO-DDN *ddname*) must be LRECL=80.

User response:

Rerun the DB2GETBACKINFO command to generate a new backinfo data set, using LRECL=80 on the DD statement. If you are unable to resolve this error, contact IBM Software Support.

CKZ07020W **Record type *type* in *ddname* file is invalid type - ignored**

Explanation:

An invalid record type was encountered in the backinfo data set.

User response:

If possible, remove the invalid record type from the backinfo data set. However, this warning can be ignored and processing continues. If you are unable to determine the cause of the warning, contact IBM Software Support.

CKZ07021E **Expected continuation record not found: *record***

Explanation:

The previous backinfo record indicated that a continuation record would follow, but no continuation record was found.

User response:

Rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07022E **Unexpected continuation record: *record***

Explanation:

An unexpected continuation record was encountered. The backinfo data set might be corrupted.

User response:

Rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07023E **No DUMPTP records found in BACKINFO file**

Explanation:

No restores can be performed because no dump tapes are defined in the backinfo data set.

User response:

Rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07024W **Volume *volume* currently is offline - will be skipped**

Explanation:

Target volume *volume* is offline and will not be used as a restore target.

User response:

Determine if the volume in question should be online or if too many offline volumes prevent the job from continuing. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07025I **Volume *volume* currently online - unit address *unit_address***

Explanation:

The volume that is listed in the message is currently online. The volume address is listed in the message.

User response:

No action is required.

CKZ07026E **ERROR CREATING COMMAND STRING; LOC=*location***

Explanation:

This is an internal error. A problem occurred in the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ07027E **BACKINFO and STATUS files are incompatible**

Explanation:

The RESTORE-FROM-DUMPTAPES command with the RERUN option was invoked, but the source volumes

entries in the STATUS file do not correspond to the source volumes entries in the BACKINFO file.

User response:

Remove the RERUN keyword (if set), delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ07032I **Volser: volser logical number of cylinders: nnnnnnnn is less than physical number of cylinders: nnnnnnnn**

Explanation:

The identified volume has a logical size (from the Format-4 DSCB) that is less than the physical size (from the DCE). The logical size will be used for pairing this volume. This may lead to a condition in which there are not enough target volumes available of the correct size to pair with all the source volumes.

User response:

No action is required. ICKDSF can be used to make the logical size equal to the physical size.

CKZ07034I **Target catalogs on target volumes**

Explanation:

There are target catalogs on the target volumes.

User response:

No action is required.

CKZ07035I **Target catalogs on target volumes**
Source catalog: source_cat_1 has volser: volser1 in the BACKINFO file. The master catalog has a different volser: volser2. The BACKINFO volser will be used

Explanation:

The target user catalog's VOLSER will be used, because the target catalog on the target volume is defined in the BACKINFO file and the master catalog has different VOLSER.

User response:

No action is required.

CKZ07036I **No target catalogs on target volumes. TARGET-UCATS-ON-TARGET-VOLUMES changed to NO**

Explanation:

The TARGET-UCATS-ON-TARGET-VOLUMES(Y) keyword is set for the RESTORE-FROM-DUMPTAPES command, but there are no ICF catalogs on the target volumes. The parameter value was switched to N.

User response:

No action is required.

CKZ07038E **Catalog: catalog on target volume: tgt_volume CATALOG IS NOT SPECIFIED AS A TARGET IN THE USERCATALOGS KEYWORD**

Explanation:

A catalog is on the target volume, but it is not specified as a target in the USERCATALOGS keyword. Processing terminates.

User response:

Correct the USERCATALOGS keyword.

CKZ07041E **program_name Unexpected results; error_text**

Explanation:

An unexpected condition occurred during the program *program_name* call. *error_text* contains a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ07042E **ERROR CKZ0900 : function: additional_message_text**

Explanation:

A parsing error was encountered on a keyword given by the *function*. If available, additional information may be provided in the *additional_message_text* field.

User response:

Contact IBM Software Support.

CKZ07043E **Internal error; Loc=location**

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ07048I **Options in effect for this execution:**

Explanation:

This message provides a summary of the runtime options after pairing volumes, resolving storage group names, and other input-related tasks.

User response:

No action is required.

CKZ07049E **User catalog user_catalog not found in backinfo_ddname file**

Explanation:

The user catalog listed in the message was not found in the backinfo data set (or the ddname given in the BACKINFO-DDN keyword).

User response:

Remove the user catalog from the USERCATALOGS keyword, or rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07050E **Keyword USERCATALOGS specified, also requires keyword USERCATALOGS-DDN (not found)**

Explanation:

The USERCATALOGS keyword was supplied, but either the UCATS ddname was not provided in the JCL, or the USERCATALOGS-DDN keyword was not supplied. One or the other is required with the USERCATALOGS keyword.

User response:

Add the UCATS DD to the JCL, or a different DD specified by USERCATALOGS-DDN, and resubmit. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07051E **ERROR - Keyword: keyword is required**

Explanation:

The keyword listed in the message is required, but it was not supplied.

User response:

Add the missing keyword and resubmit.

CKZ07052E **ERROR - Keyword: keyword has odd number of values, must be even number of paired values**

Explanation:

The keyword listed in the message has an odd number of values. The number of values must be even.

User response:

Correct the input and resubmit.

CKZ07053E **ERROR - Keyword: keyword error_text additional_message_text**

Explanation

The keyword that is listed in the message has an error. The error is described in the message text. Possible values for *error_text* are:

- value too long, max length = xx
- value must be numeric
- value must be 1-16 inclusive

More information might be provided in the *additional_message_text* field.

User response:

Correct the input and resubmit.

CKZ07055E **Keyword USERCATALOGS specified but no UCAT records found in ddname file**

Explanation:

The USERCATALOGS keyword requires the same user catalogs to be defined in the backinfo data set (identified in the message by *ddname*).

User response:

Rerun the DB2GETBACKINFO command with the USERCATALOGS keyword and the correct user catalogs to regenerate the backinfo data set. Use the new backinfo data set and rerun RESTORE-FROM-DUMPTAPES. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07057E **DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry**

Explanation:

The indicated entry for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ07058E **Invalid value in keyword: keyword value: value**

Explanation:

The keyword and its value that are listed in message are invalid.

User response:

Refer to the documentation for valid values for this command and keyword. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07060E **UCBLOOK error; return code=return_code reason code=reason_code LOC=location**

Explanation:

The UCBLOOK macro failed with the given return code and reason code and at the given location (for internal use only).

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07061E **CKZ01SMF error; return code=return_code loc =location additional_message_text**

Explanation:

Storage group lookup failed with the give return code that is listed in the message, at the given location (for internal use only). More information might be provided in the *additional_message_text* field; also refer to accompanying message CKZ07062E.

User response:

Refer to message CKZ07062E for the name of the storage group that returned the error. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07062E **No volumes returned by SSI for storage group: *storage_group***

Explanation:

This message is used with CKZ07061E, and indicates that storage group lookup failed for the storage group that is listed in the message. No volumes were found.

User response:

Verify the storage group name. If necessary, consult your DASD administrators. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07063E **Maximum volumes reached = 9999**

Explanation:

The maximum number of volumes for one command was exceeded.

User response:

Reduce the number of volumes for one command by dividing the backinfo data set into separate files and using multiple RESTORE-FROM-DUMPTAPES commands. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07066E **NO SOURCE VOLSER WITH CATALOG FOUND FOR TARGET VOLSER: *volser_additional_text***

Explanation

An attempt to find a source volume that has a catalog on it for the target volume that has a catalog failed. All of the source volumes that were rejected are printed after this message, along with.

No suitable source volume was found for a target volume based on the target volume's device type, capability, storage subsystem, number of tracks, and sequence number (if KEEP-VOLUMES-SEQUENCE(Y) is specified). Processing terminates.

The *additional_text* message string contains the following information:

- For target volume data:

```
volser SIZE=size STOGROUP=stogroup
SRC.SEQ.STOGROUP=src_stogroup CATALOG=catalog
```

- For source volume data:

```
volser SIZE=size STOGROUP=stogroup
CATALOG=catalog reason_for_rejection
```

where:

volser

Source or target volume serial.

size

The size of the volume in tracks.

stogroup

storage group name of the target or the source volume. For target volumes, this value is printed when this volume was defined by TO-STORAGEGROUP. For source volumes, this value is printed when this volume was defined by SOURCE-STORAGEGROUP.

src.stogroup

(Target volumes only) The source storage group name of the target volume. This value is printed when the target volume was defined by TO-STORAGEGROUP. The SOURCE-STORAGEGROUP and the position of *src.stogroup* is same as the position of *stogroup* in TO-STORAGEGROUP.

catalog

The target catalog that is placed on the target volume.

reason_for_rejection

Can be one of the following:

PAIRED WITH *tgt_volser*: The source volume is already paired with another target volume *tgt_volser*.

FEWER TRACKS: The target volume is smaller than the corresponding source volume.

DIFFERENT STORAGE GROUPS: The KEEP-VOLUMES-SEQUENCE(Y) keyword was specified, but the corresponding source volume has a different position in the SOURCE-STORAGEGROUP keyword than the target volume in the TO-STORAGEGROUP keyword; or, the SOURCE-STORAGEGROUP or TO-STORAGEGROUP keywords were not specified.

User response:

Ensure that for every target volume that has a catalog on it, there is a source volume with catalog on it that meets the required criteria. If VOLSER masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ07067E **No volumes in storage group(s)**

Explanation:

None of the specified target storage groups have volumes assigned to them.

User response:

Correct the target storage group names, or add volumes to the target storage groups. If you are unable

to determine the cause of this error, contact IBM Software Support.

CKZ07071E **More source volsers
number_of_source_volsers than
online target volsers
number_of_target_volsers**

Explanation:

The number of source volumes exceeds the number of online target volumes.

User response:

Add more target volumes or storage groups to the target list, or bring more target volumes online. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07072E **More source volsers
number_of_source_volsers than
target volsers
number_of_target_volsers**

Explanation:

The number of source volumes exceeds the number of target volumes.

User response:

Add more target volumes or storage groups to the target list. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07073E **Volume matching error for target
volume target_volser
additional_text**

Explanation

The target volume listed in the message could not be matched with any source volume or dumptape. Possible reasons include: the volume is part of a storage group, but the source storage group that it is matched with is empty; or all remaining source volumes are too large to match with this target volume.

The *additional_text* message string contains the following information:

- The volume size in cylinders and the storage group name of the source volume.
- A table that contains information about all of the target volumes is output, one row for each target volume. The table contains the following columns:

Target

The target volume serial number.

Size

The size of the target volume.

Sgname

The storage group name of the target volume. If it is not defined, then this value is blank.

Reason

The reason for the target volume's rejection. This can be one of the following:

Paired with tgt_volser

The source volume is already paired with another target volume *tgt_volser*.

Size too small

The target volume is smaller than the corresponding source volume;

Different seq.

The KEEP-VOLUMES-SEQUENCE(Y) keyword was specified, but the corresponding source volume has a different position in SOURCE-STORAGEGROUP than the target volume in TO-STORAGEGROUP, or the SOURCE-STORAGEGROUP or TO-STORAGEGROUP parameters were not specified.

Unknown error

An unknown error occurred.

User response:

Investigate the potential cause based on the suggestions above. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07076I **Volume target_volser paired with
source volume source_volser**

Explanation:

This informational message lists the target volume and source volume pair.

User response:

No action is required.

CKZ07077E **User catalog lookup error
error_text**

Explanation:

The catalog search interface failed for the reason given in *error_text*.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07078E **User catalog user_catalog resides
on target volume target_volume**

Explanation:

Target user catalogs may not reside on target volumes.

User response:

Move the user catalog to a different volume before rerunning the RESTORE-FROM-DUMPTAPES command. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07087E **Masking characters not allowed
with keyword keyword**

Explanation:

Masking characters are not supported with the keyword that is listed in the message.

User response:

Use specific values for this keyword. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07089E Limit of 32,000 wildcard volumes exceeded

Explanation:

No single wildcard mask for volume names can exceed 32,000 volumes. The overall limit for all volumes derived from all wildcard masks is also 32,000.

User response:

Use wildcard masks that return a smaller number of volumes. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07090I Volume *volser* included via volume mask *volume_mask*

Explanation:

The *volser* that is listed in the message is included because it matches the given volume mask.

User response:

No action is required.

CKZ07091E Error getting unit address for volume *volser* from STATUS file

Explanation:

The RESTORE-FROM-DUMPTAPES command with RERUN option was invoked, but the entry about the target volume with status='R' in the STATUS file does not contain information about the unit address of this volume

User response:

Ensure that the target volume *volser* is not the same as the *volser* of the source volume, and that the target *volser* is online. If the target volume *volser* does not match the specified *volser*, or the target volume is offline, then restore the *volser* of the target volume and/or bring the target volume online. Then resubmit the RESTORE-FROM-DUMPTAPES command without the RERUN option. If you are unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ07092E Target volume with serial # *volser* must be unavailable because it was restored but not clipped/online

Explanation:

The RESTORE-FROM-DUMPTAPES command with the RERUN option was invoked, but the entry for the target volume information with status='R' is in the STATUS file, and the volume is available in the system.

User response:

Remove the RERUN keyword (if set), delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ07093E Target volume *volser* was restored in previous run but now it is not selected

Explanation:

The RESTORE-FROM-DUMPTAPES command with the RERUN option was invoked, but the target volume was restored or clip was completed in accordance with the STATUS file and the volume was not selected in this command.

User response:

Check the selected volumes in both the RESTORE-FROM-DUMPTAPES command without RERUN and the RESTORE-FROM-DUMPTAPES command with RERUN. If the set of volumes in the command without RERUN is incorrect, then correct the set of target volumes and repeat the run of RESTORE-FROM-DUMPTAPES command without the RERUN option. If the set of volumes in the command with RERUN is incorrect, then correct the set of target volumes and repeat the run of RESTORE-FROM-DUMPTAPES command with the RERUN option. If you are unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ07094E STATUS file restore records not found - prevents save of VOLPAIRS

Explanation:

There is no correct information in the STATUS file.

User response:

Remove the RERUN keyword (if set), delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ07095E Error accessing STATUS file; *loc=location*

Explanation:

A STATUS file open failed at the given location when data was being written to the VOLPAIRS file (for internal use only).

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ07096E No records in STATUS file

Explanation:

A STATUS file does not contain information about pairs of source and target volumes before writing to the VOLPAIRS file (for internal use only).

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ07102I **End of CKZ00071 processing –
max rc return_code**

Explanation:

Program CKZ00071 is complete and returned the return code that is listed in the message.

User response:

No action is required.

CKZ07104E **LOAD failed for module:
module_name**

Explanation:

An attempt to load the program that is listed in the message text failed.

User response:

Ensure that the Db2 Cloning Tool runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07105E **Allocation failed for ddname:
ddname**

Explanation:

An attempt to allocate the file that is listed in the message text failed.

User response:

Ensure that the Db2 Cloning Tool runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07106W **Deallocation failed for ddname:
ddname**

Explanation:

An attempt to deallocate the file that is listed in the message text failed.

User response:

This warning might not be a problem. If you are unable to determine the cause of this message, contact IBM Software Support.

CKZ07107E **ATTACHX failed RC return_code
additional_message_text**

Explanation:

An attempt to attach a subtask failed with the return code that is listed in the message.

User response:

Ensure that the Db2 Cloning Tool runtime libraries are included in the STEPLIB concatenation. If you are

unable to determine the cause of this error, contact IBM Software Support.

CKZ07111E **Error accessing STATUS file;
loc=location**

Explanation:

An attempt to insert, read with update, or rewrite a record to the STATUS file failed.

User response:

Save the job listing and the STATUS file and contact IBM Software Support.

CKZ07112E **Duplicate STATUS file entry;
loc=location**

Explanation:

An attempt to insert a record into the STATUS file failed.

User response:

Save the job listing and the STATUS file and contact IBM Software Support.

CKZ07114E **STATUS file restore record not
found**

Explanation:

An attempt to read-with-update a record in the STATUS file failed because the record was not found.

User response:

Save the job listing and the STATUS file and contact IBM Software Support.

CKZ07118E **ERROR: failure condition_code at
module_name + offset - *****
TERMINATING *******

Explanation:

An unrecoverable program error occurred. The completion code, module name and offset are listed in the message text.

User response:

Save the job listing and the STATUS file and contact IBM Software Support.

CKZ07119E **Unable to establish ESTATEX;
R15= return_code**

Explanation:

An attempt to establish error recovery via the ESTAEX macro failed with the return code listed in the message text.

User response:

Save the job listing and the STATUS file and contact IBM Software Support.

CKZ07120W **Skipping clip / vary for volume
volser due to prior restore task
error**

Explanation:

Failure of the restore task prevents further processing of the volume that is listed in the message.

User response:

Resolve the cause of the restore task error before proceeding. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07121E Fatal error with clip / vary on volume *volser* – exiting

Explanation:

An essential part of the post-restore processing for the volume that is listed in the message failed.

User response:

Review other associated messages to determine and fix the error. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07125E <<>> Found *number* offline target volumes - TERMINATING

Explanation:

There are not enough target volumes available online to continue. The number of offline volumes from the pool of target volumes is listed in the message. The command terminates with a return code of 8.

User response:

Specify a different set of target volumes, or bring more target volumes online, or both.

CKZ07130I Restore task completed: target *volser* tape *tape_volume* file *file_sequence_number* return code *return_code*

Explanation:

This message indicates that a single restore for the target volume and tape volume listed in the message completed with return code of *return_code*.

User response:

If the return code is nonzero, review other applicable messages. Otherwise, no action is required.

CKZ07131E Remaining restores incomplete:

Explanation:

One or more errors occurred that leave some restores incomplete. Details are provided in the accompanying CKZ07132I messages.

User response:

Refer to other messages that are related to this error in the job log or Db2 Cloning Tool message file. Rerun the job with appropriate changes. If you are unable to resolve this error, have the job listing and associated files available, and contact IBM Software Support.

CKZ07132I *target_volser* requiring tape *tape_volume* *additional_information*

Explanation:

This message accompanies message CKZ07131E and provides details for a target volume that did not get restored and the tape volumes that it requires. Additional information may be provided in the message.

User response:

No action is required.

CKZ07140E Unexpected condition: *error_text*

Explanation:

An unexpected condition was encountered.

User response:

Have the job listing and associated files available, and contact IBM Software Support.

CKZ07160E UCBLOOK error: return code=*return_code* reason code=*reason_code* LOC=*location*

Explanation:

The UCBLOOK macro failed. The return and reason codes are listed in the message, as well as the internal location given by *location*.

User response:

Have the job listing and associated files available, and contact IBM Software Support.

CKZ07201I Restore started - program *rev=revision* [** SIMULATION **]

Explanation:

The restore driver for one target volume began execution. The program revision is listed in the message.

User response:

No action is required.

CKZ07202I Restore completed; return code=*return_code*

Explanation:

The restore driver for one target volume completed with the return code that is listed in the message.

User response:

No action is required.

CKZ07203I DDname=*ddname* allocated for dsn=*data_set_name*

Explanation:

The ddname that is listed in the message has been allocated for the data set name *data_set_name*. For tape allocations, *data_set_name* display as **TAPE**, followed by the tape volume and file sequence number.

User response:

No action is required.

CKZ07204E Open failed for ddname=ddname**Explanation:**

An attempt to open the data set with ddname *ddname* failed.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07205E Allocation failed for
 dsn=data_set_name****Explanation:**

An attempt to allocate the data set that is listed in the message failed.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07206W Deallocation failed for ddname:
 ddname****Explanation:**

An attempt to deallocate the data set for the ddname that is listed in the message failed.

User response:

This message is a warning and may not affect the restore process. If you are unable to determine the cause of this warning, contact IBM Software Support.

**CKZ07209I Allocation failed for
 dsn=data_set_name Will attempt
 to allocate it within nnn minutes.****Explanation:**

An attempt to allocate the data set *data_set_name* failed. Allocation attempts for this data set will be repeated every 10 seconds for the next *nnn* minutes.

User response:

No action is required.

**CKZ07210E All allocation attempts for
 dsn=data_set_name had failed.
 Wait time=nnn, return
 code=return_code.****Explanation:**

All allocation attempts for the data set *data_set_name* within the last *nnn* minutes failed. Processing terminates.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07214I Tape *tape_volume*, fileseq
 file_sequence_number, dsn
 data_set_name ** SIMULATION
 ******Explanation:**

This message displays in simulation mode and lists the tape volume, file sequence number, and data set name that would be allocated for this restore during a non-simulation run.

User response:

No action is required.

**CKZ07221E ADRDSSU restore failed; R15:
 return_code****Explanation:**

The restore program ADRDSSU failed with the return code that is listed in the message.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07301I Clip target started - program rev=
 revision****Explanation:**

The process of clipping a target volume started. The revision number of the program is shown in the message.

User response:

No action is required.

**CKZ07302I Clip target volume *volume devn*
 device_number completed; return
 code= return_code****Explanation:**

The process of clipping a target volume completed with the listed *return_code*. The information for the volume that was clipped is shown in the message text.

User response:

No action is required.

**CKZ07303I ddname *ddname* allocated for dsn
 *data_set_name*****Explanation:**

The data set that is listed in the message was successfully allocated with ddname *ddname*.

User response:

No action is required.

CKZ07304E Open failed for ddname *ddname***Explanation:**

The ddname that is listed in the message cannot be opened.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07305E Allocation failed for dsn
 *data_set_name*****Explanation:**

The data set that is listed in the message could not be allocated.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07306W Deallocation failed for ddname
 ddname**

Explanation:

The ddname that is listed in the message could not be deallocated.

User response:

If you are unable to determine the cause of this warning, contact IBM Software Support.

**CKZ07308E Unable to load program:
 program_name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Ensure that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ07322E ERROR CREATING COMMAND
 STRING; LOC=location**

Explanation:

This is an internal error. A problem occurred in the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ07335E ICKDSF command failed for target
 volume volume**

Explanation:

An attempt to relabel the target volume that is listed in the message failed.

User response:

If you are unable to determine the cause of this warning, contact IBM Software Support.

**CKZ07336E IEEVARYD vary on/offline failed
 for target volume volume; invalid
 parameters**

Explanation:

An attempt to vary the target volume that is listed in the message online or offline failed due to invalid parameters.

User response:

Contact IBM Software Support.

**CKZ07337W IEEVARYD vary on/offline failed
 for target volume volume device
 device return code return_code
 reason code reason_code**

Explanation:

An attempt to vary the target volume that is listed in the message online or offline failed due to invalid parameters. Details about the target volume are provided in the message text.

User response:

Contact IBM Software Support.

**CKZ07338W IEEVARYD vary on/offline failed
 for target volume volume device
 device R15 return_code**

Explanation:

An attempt to vary the target volume that is listed in the message online or offline failed. The device information and return code are provided in the message.

User response:

If you are unable to determine the cause of this warning, contact IBM Software Support.

**CKZ07339E IEEVARYD vary offline failed for
 target volume volume device
 device – retry loop exhausted**

Explanation:

An attempt to vary the target volume that is listed in the message offline failed until the retry loop is exhausted.

User response:

If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ07340I Device number device successfully
 changed to volume serial volser**

Explanation:

The device was successfully relabeled with the volume serial that is listed in the message.

User response:

No action is required.

**CKZ07341I Volume serial volser device
 number device is now online**

Explanation:

The volume that is listed in the message is now online.

User response:

No action is required.

**CKZ07348I Volume serial volser_1 device
 number device not clipped to
 volume serial volser_2 - **
 SIMULATION ****

Explanation:

volser_1 at the device number *device* was not relabeled to volume *volser_2* because the command was run in SIMULATION mode.

User response:

No action is required.

CKZ10001I **hh:mm:ss RENAME PROCESS
STARTED - PROGRAM REV=rrr |
hh:mm:ss RENAME PROCESS
COMPLETED; RETURN CODE=nnn**

Explanation:

RENAME command processing message.

User response:

No action is required.

CKZ10003I **DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ10004E **DDNAME MISSING: ddname |
OPEN FAILED FOR DDNAME:
ddname**

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ10005E **ALLOCATION FAILED FOR DSN:
datasetname | ALLOCATION
FAILED FOR DDNAME: ddname**

Explanation:

Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ10005W **DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ10006E **ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ10007W **ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ10008E **UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ10009E **ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ10010E **DUPLICATE JOURNAL ENTRY;
LOC=lllll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ10011E **JOURNAL CONTROL RECORD NOT FOUND | JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL USER CATALOG RECORD(S) NOT FOUND | JOURNAL VOLUME PAIR RECORD(S) NOT FOUND | JOURNAL EXCLUDE RENAME MASK RECORD(S) NOT FOUND | JOURNAL TARGET CATALOG RECORD(S) NOT FOUND | JOURNAL TARGET CATALOG RECORD IS WRONG VERSION**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file or has the wrong version. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ10013E **RECORD COUNT IS ZERO; LOC=lllll | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc**

Explanation:

There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ10015E **THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY**

Explanation:

The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response:

Check that the COPY command has completed successfully before initiating the RENAME command.

CKZ10016I **COPY WAS A SIMULATION; RENAME CHANGED TO SIMULATION**

Explanation:

The journal indicates that the COPY command was a simulation. Processing continues, but, the RENAME will be run as a simulation.

User response:

No action is required.

CKZ10017E **THE DDNAME IS EMPTY OR HAS BEEN DUMMIED, DDNAME: ddn**

Explanation:

No records were read from the ddname specified for a keyword. Processing terminates.

User response:

Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ10019E **THE keyword DOES NOT HAVE A LRECL OF 80, DDNAME: ddn**

Explanation:

The data set allocated to the ddname for the identified keyword does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response:

Change the data set allocated to the ddname to have a LRECL of 80.

CKZ10020E **SOURCE CATALOG BACKUP HAS NOT BEEN DONE**

Explanation:

The source catalogs have not been backed up. Processing terminates.

User response:

Run UCATOPTIONS BACKUP to backup the source catalogs.

CKZ10039E **INTERNAL ERROR: *error_text***

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ10040E *multiple possible messages; see Explanation*

Explanation

ERROR ATTACHING DRIVER, PGM=program name

The indicated program name was not found. Processing terminates.

UNEXPECTED CONDITION, TASK NOT POSTED

The program returned from a 'WAIT', but, had not been posted. Processing terminates.

User response:

For ERROR ATTACHING DRIVER message, check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support. For UNEXPECTED CONDITION message, contact IBM Software Support. Have available the listing that contains this message.

CKZ10041E CKZ00900 UNEXPECTED RESULTS; error text

Explanation:

An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ10042E RERUN NOT SPECIFIED AND PREVIOUS RUN WAS NOT A SIMULATION

Explanation:

A RENAME was attempted without the RERUN parameter, but the journal indicates that a non-simulation RENAME has already been attempted.

User response:

If 'SAFE' mode was used for the first RENAME, specify 'RERUN' for the RENAME command.

CKZ10043E VOLBKUP DSNNAME MISMATCH: CURRENT DSN=current dsn in command PREVIOUS DSN=previous dsn used

Explanation:

A RENAME with RERUN is pointing to a different data set for the volume backup file.

User response:

Correct the data set name for the volume backup file in the JCL.

CKZ10044E RERUN WAS SPECIFIED AND PREVIOUS RUN WAS NOT SAFE

Explanation:

A RENAME RERUN was attempted, but the journal indicates that a previous RENAME did not include the SAFE parameter.

User response:

The COPY command will need to be run before initiating the RENAME command without the RERUN parameter.

CKZ10048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how RENAME will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ10050E ERROR IN PARAMETERS FOR keyword

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ10051E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ10052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

CKZ10053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ10054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ10056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ10057E **DUPLICATE FOUND; KEYWORD:**
keyword ENTRY: entry

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ10058E **INVALID VALUE IN KEYWORD:**
keyword VALUE: value error text

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ10059I **TARGET CATALOGS ON TARGET**
VOLUMES

Explanation:

There are target catalogs on the target volumes.

User response:

No action is required.

CKZ10060I **RENAME-MASKS KEYWORD**
ENTRY FOR SOURCE CATALOG
HAS BEEN REPLACED; DSN:
dsnmask

Explanation:

A fully qualified entry already exists for this source catalog entry. This entry is replaced with the new one.

User response:

No action is required.

CKZ10061E **CKZ01SMF ERROR; RETURN**
CODE=nnnn LOC: llllll INVALID
smstypeCLAS NAME: class entry

Explanation:

An error occurred using CKZ01SMF to verify the SMS class specified. llllll is the internal location where the error occurred. smstype indicates DATA, MGMT, or, STOR. Processing terminates.

User response:

Correct the indicated SMS class entry.

CKZ10068E **UNPAIRED ENTRIES IN**
KEYWORD: keyword

Explanation:

The keyword requires pairs of entries. An odd number of entries was found. Processing terminates.

User response:

Correct the keyword specification.

CKZ10069E **SMF IS NOT RECORDING THE**
SPECIFIED RECORD TYPE: nnn

Explanation:

The SMF audit log has been requested but SMF is not recording the specified record type. Processing terminates.

User response:

Correct the keyword specification to use a record type that SMF is recording or have SMF record the specified record type.

CKZ10085I **DSNS FOR KEYWORD: keyword**
PROCESSING SEQUENCE list of
dsns number

Explanation:

Parsing found the listed dsns for the keyword. The processing sequence number shows the order that the dsns were entered and will be the order used during processing.

User response:

No action is required.

CKZ10086I **VALIDATING KEYWORD: keyword**

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

CKZ11001I **hh:mm:ss VOLUME UPDATES**
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss
VOLUME UPDATES COMPLETED;
RETURN CODE=nnn

Explanation:

VOLUME UPDATES processing message.

User response:

No action is required.

CKZ11004E **OPEN FAILED FOR**
DDNAME=ddname

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ11007W **ERROR CALLING CKZ01HEX;**
FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ11008E UNABLE TO LOAD PROGRAM:
program_name

Explanation:

The indicated program name was not found.
Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ11009E ERROR ACCESSING JOURNAL
FILE; LOC=lllll

Explanation:

A VSAM error occurred accessing the journal file.
Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ11010E COMMAND FAILED: *command* IN
MODULE CKZ00047

Explanation:

An error occurred while calling CKZ00047. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11011E JOURNAL VOLUME PAIR RECORD
**NOT FOUND FOR source volume/
target volume**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11024E VOLUME BACKUP FAILED FOR
volser. SEE SYSOUT DRSTATS FOR
MORE INFORMATION

Explanation:

An error occurred during SAFE mode attempting to back up a volume. Processing terminates.

User response:

Check the DRSTATS for the error that occurred. If unable to correct the problem, contact IBM Software Support.

CKZ11030I VOLUME CONVERSION STARTED
FOR source volume/target volume

| VOLUME CONVERSION

**COMPLETED FOR source volume/
target volume; RETURN CODE=nn**
DATA SETS=nnnnnn | VOLUME
CONVERSION FAILED FOR source
volume/target volume; RETURN
CODE=nn DATA SETS=nnnnnn

Explanation:

Volume processing information message. DATA SETS=nnnnnn indicates the number of format 1 dscb's renamed on the target volume.

User response:

None, unless the message includes the word FAILED or the RETURN CODE= is greater than four (4). If so, check the sysout for that target volume for the error messages.

CKZ11031I VOLUME CONVERSION TOTAL
DATA SETS: nnnnnnnn

Explanation:

Volume processing information message. The number is the total number of format 1 dscb's renamed on all target volumes.

User response:

No action is required.

CKZ11032I VOLUME BACKUP COMPLETED
FOR target volume

Explanation:

Volume processing information message. During 'SAFE' mode, the target volume's VTOC, VTOCIX, and VVDS are backed up before any changes occur.

User response:

No action is required.

CKZ11033W VOLUME CONVERSION BYPASSED
FOR source volume/target volume
DUE TO PRIOR ERRORS

Explanation:

Volume processing has not been done for the target volume due to errors that happened while processing another volume. The other volumes with errors can be determined from the prior messages: CKZ11030I VOLUME CONVERSION FAILED FOR srcvol/tgtvol

User response:

Correct the cause for the errors that happened to the other volumes and do a RENAME RERUN if possible. If RENAME RERUN is not possible the COPY will have to be redone.

CKZ11034I TOTAL BYTES WRITTEN TO
VOLBKUP DDNAME=ddname;
nnnnnnnn

Explanation:

Volume processing information message. This shows the number of bytes that were written to the VOLBKUP file.

User response:
No action is required.

CKZ11035I **TARGET VOLUME volser IS AN EXTENDED ADDRESS VOLUME; ICKDSF WILL BE USED TO REBUILD THE VTOCIX**

Explanation:
The identified volume is an Extended Address Volume and VTOCIX_REBUILDER = MSC has been specified in the INI. For Extended Address Volumes ICKDSF will always be used to rebuild the VTOCIX.

User response:
No action is required.

CKZ11036E **RECORD LENGTH FOR VOLUME BACKUP IS TOO BIG | RECORD LENGTH FOR VOLUME BACKUP IS TOO SMALL | RECORD LENGTH FOR VOLUME BACKUP IS ZERO**

Explanation:
The size that was retrieved from the target volume for BACKUP is too small, too big, or 0.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ11040E **UNEXPECTED CONDITION; error text**

Explanation:
An unexpected condition occurred while dispatching a volume task or while waiting for the completion of a volume task. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ11041E **TARGET VOLUME target volume IS NOT IN A PROPER STATE FOR RENAME**

Explanation:
The target volume is not in a proper state for RENAME processing. The volume may have been modified between COPY and RENAME which is not allowed. Processing terminates.

User response:
The COPY will need to be run again to put the target volume into the proper state for RENAME processing.

CKZ11042W **UNEXPECTED CONDITION; error text**

Explanation:

An unexpected condition occurred while dispatching a service task. Processing continues.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ11060E **ERROR DURING function FOR VOLSER=volume - RETURN CODE=nnnn REASON CODE=nnnn**

Explanation:
An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response:
A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11089I **NOT ALL DATASETS HAVE BEEN RENAMED ON THE VOLUMES**

Explanation:
An error occurred during volume rename processing that caused some data sets to not be renamed on a volume. Message CKZ11030I will indicate which volume pair had an error.

User response:
Correct the cause of the error and rerun RENAME if possible.

CKZ11101I **hh:mm:ss VOLUME UPDATE STARTED - PROGRAM REV=rrr (** SIMULATION **) CKZ11101I hh:mm:ss VOLUME UPDATE COMPLETED; RETURN CODE=nnn F1DSCB COUNT=nnnnnnnn**

Explanation:
Individual VOLUME UPDATE processing message.

User response:
No action is required.

CKZ11102I **RENAMING VTOC ENTRIES ON VOLUME: target volume (** SIMULATION **) | RENAMING VTOCIX ENTRIES ON VOLUME: target volume (** SIMULATION **)**

Explanation:
Individual VOLUME UPDATE processing messages.

User response:
No action is required.

CKZ11104E **OPEN FAILED FOR DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11105E ALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic allocation for a ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ11109E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ11110W VOLSER IN FORMAT 1 NOT IN
VOLSER PAIRS: volser found DSN:
datasetname**

Explanation:

The volume serial found in the format 1 dscb for the indicated data set is not in the volume pairs. Processing continues.

User response:

This probably is a condition copied from the source volumes. If desired, correct the format 1 dscb on the source volume for the data set. (An incorrect volume serial in the format 1 dscb will not prevent accessing the data set.)

**CKZ11112W DATA SET MATCHES NO RENAME
MASK: datasetname (MATCHES
EXCLUDE MASK)**

Explanation:

The indicated data set did not match any source rename mask. If the message includes (MATCHES EXCLUDE MASK) the data set matched an entry in the EXCLUDE-SRCNAME-MASKS keyword. Processing continues.

User response:

No action is required.

**CKZ11114E UNEXPECTED VALUE FOUND IN
VVDS**

Explanation:

During rename of the SYS1.VVDS data set, something unexpected was found in the self-defining entry. The entry is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11115E VOLUME VOLSER NOT EQUAL TO
NEW VOLSER: volser in label**

Explanation:

During volume processing, the volume serial in the label does not match the target volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11117E RENAME WOULD CAUSE A
DUPLICATE DSN IN VTOC: source
dsn NEWNAME: target dsn**

Explanation:

The target dsn would result in a duplicate name in the VTOC. Processing terminates.

User response:

Correct the RENAME-MASKS to prevent the creation of duplicate data set names.

**CKZ11120E UNPOPULATED SECTION; VIR
ENTRIES AVAILABLE**

Explanation:

An internal error occurred during VTOCIX processing.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11121E EXPECTED VIXM ENTRY NOT
FOUND**

Explanation:

An internal error occurred during VTOCIX processing.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11122E EXPECTED VIB ENTRY NOT FOUND

Explanation:

An internal error occurred during VTOCIX processing.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11123E UNKNOWN RECORD TYPE FOUND
IN VTOCIX**

Explanation:

An internal error occurred during VTOCIX processing.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11124E UNABLE TO VALIDATE DCE FOR
VOLUME: target volume**

Explanation:

The DCE fields for the target volume have not been set. Accessing the VTOC did not cause the DCE fields to be corrected.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11130E RETURN CODE 8 SET FOR
TEMPORARY DATA SET(S)**

Explanation:

A return code of 8 was requested for the indicated condition. Processing terminates.

User response:

No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ11131I RENAMED DATA SETS:

Explanation:

A RENAME SIMULATE was requested. The data set names that are listed will be renamed on the target volume.

User response:

No action is required.

**CKZ11133E EXCP FAILED: function | SYNAD
TEXT: error text from SYNADAF**

Explanation:

An EXCP request failed. SYNADAF was invoked to help diagnose the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11143E UNABLE TO RENAME DSN=source
datasetname | USING
MASK=target mask**

Explanation:

The new name of a data set will exceed 44 characters. Processing terminates.

User response:

Correct the RENAME-MASKS specification.

**CKZ11157E type VOLSER: volser IS EXTENDED
ADDRESS VOLUME**

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support.

**CKZ11160E ERROR DURING function FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ11161E ERROR CALLING SMFWTM;
R15=nnnn**

Explanation:

An error occurred using the SMFWTM macro. Processing terminates.

User response:

If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11201I hh:mm:ss **VOLUME UPDATE STARTED - PROGRAM REV=rrr (** SIMULATION **) | hh:mm:ss VOLUME UPDATE COMPLETED; RETURN CODE=nnn F1DSCB COUNT=nnnnnnnn F8DSCB COUNT=nnnnnnnn**

Explanation:

Individual VOLUME UPDATE processing message.

User response:

No action is required.

CKZ11202I **RENAMING VTOC ENTRIES ON VOLUME: target volume (** SIMULATION **)**

Explanation:

Individual VOLUME UPDATE processing message.

User response:

No action is required.

CKZ11203I **DDNAME=ddname ALLOCATED FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ11204E **OPEN FAILED FOR DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ11205E **ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11205W **DEALLOCATION FAILED FOR DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11206E **ERROR CALLING CKZ01VV2 tablename FUNCTION: function R15=register15 R0=register0 LOC=location**

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ11207W **ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ11209E **ERROR ACCESSING JOURNAL FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ11210W **VOLSER IN FORMAT 1 NOT IN VOLSER PAIRS: volser found DSN: datasetname**

Explanation:

The volume serial found in the format 1 dscb for the indicated data set is not in the volume pairs. Processing continues.

User response:

This probably is a condition copied from the source volumes. If desired, correct the format 1 dscb on the source volume for the data set. (An incorrect volume serial in the format 1 dscb will not prevent accessing the data set.)

CKZ11212W DATA SET MATCHES NO RENAME MASK: datasetname (MATCHES EXCLUDE MASK)

Explanation:

The indicated data set did not match any source rename mask. If the message includes (MATCHES EXCLUDE MASK) the data set matched an entry in the EXCLUDE-SRCNAME-MASKS keyword. Processing continues.

User response:

No action is required.

CKZ11214E UNEXPECTED VALUE FOUND IN VVDS

Explanation:

During rename of the SYS1.VVDS data set, something unexpected was found in the self-defining entry. The entry is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11215E VOLUME VOLSER NOT EQUAL TO NEW VOLSER: volser in label

Explanation:

During volume processing, the volume serial in the label does not match the target volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11217E RENAME WOULD CAUSE A DUPLICATE DSN IN VTOC: source dsn NEWNAME: target dsn

Explanation:

The target dsn would result in a duplicate name in the VTOC. Processing terminates.

User response:

Correct the RENAME-MASKS to prevent the creation of duplicate data set names.

CKZ11230E RETURN CODE 8 SET FOR NOT RENAMED DATA SET(S) | RETURN CODE 8 SET FOR TEMPORARY DATA SET(S)

Explanation:

A return code of 8 was requested for the indicated condition. Processing terminates.

User response:

No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member,, or override the return code in the RENAME command.

CKZ11231I RENAMED DATA SETS:

Explanation:

A RENAME SIMULATE was requested. The data set names that are listed will be renamed on the target volume.

User response:

No action is required.

CKZ11233E SYNAD TEXT: error text from SYNADAF

Explanation:

An EXCP request failed. SYNADAF was invoked to help diagnose the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11235W ICKDSF COMMAND FAILED FOR TARGET VOLSER: target volser

Explanation:

The ICKDSF command failed. The messages from ICKDSF will be printed.

User response:

If unable to determine the reason for the ICKDSF failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ11243E UNABLE TO RENAME DSN=source datasetname | USING MASK=target mask

Explanation:

The new name of a data set will exceed 44 characters. Processing terminates.

User response:

Correct the RENAME-MASKS specification.

CKZ11260E ERROR DURING function FOR VOLSER=volume - RETURN CODE=nnnn REASON CODE=nnnn

Explanation:

An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11261E ERROR CALLING SMFWTM; R15=nnnn

Explanation:

An error occurred using the SMFWTM macro. Processing terminates.

User response:

If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11301I **hh:mm:ss VVDS UPDATE STARTED FOR VOLUME: volume - PROGRAM REV=rrr (** SIMULATION **) | hh:mm:ss VVDS UPDATE COMPLETED; RETURN CODE=nnn**

Explanation:

Individual VVDS UPDATE processing message.

User response:

No action is required.

CKZ11303I **DDNAME=ddname ALLOCATED FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ11305E **ALLOCATION FAILED FOR DSN: datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11305W **DEALLOCATION FAILED FOR DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11306E **ERROR CALLING CKZ01VV2 tablename FUNCTION: function R15=register15 R0=register0 LOC=location**

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ11307W **ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ11309E **ERROR ACCESSING JOURNAL FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ11312W **DATA SET MATCHES NO RENAME MASK: datasetname (MATCHES EXCLUDE MASK)**

Explanation:

The indicated data set did not match any source rename mask. If the message includes (MATCHES EXCLUDE MASK) the data set matched an entry in the EXCLUDE-SRCNAME-MASKS keyword. Processing continues.

User response:

No action is required.

CKZ11322E **ERROR CREATING COMMAND STRING; LOC=location**

Explanation:

This is an internal error. A problem occurred in the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11330E **RETURN CODE 8 SET FOR NOT RENAMED DATA SET(S) | RETURN CODE 8 SET FOR MISSING USER CATALOG(S)**

Explanation:

A return code of 8 was requested for the indicated condition. Processing terminates.

User response:

No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ11335I SMS smstypeCLAS COPIED FROM SOURCE FOR VVDS ENTRY - dsname

Explanation:

No default value was given for the smstype (DATA, MGMT, or, STOR). The indicated SMS class for the entry was copied from the source data set.

User response:

No action is required, unless a specific class is desired.

CKZ11340W RENAMES NOT COMPLETE FOR VVDS ENTRY - COMPONENT NAME=dsname

Explanation:

A VVDS entry could not be completely renamed. Processing continues. The entry will be handled as a 'not renamed' data set.

User response:

No action is required.

CKZ11341W RENAMES NOT DONE FOR VVDS ENTRY - COMPONENT NAME=dsname

Explanation:

A VVDS entry could not be renamed. Processing continues. The entry will be handled as a 'not renamed' data set.

User response:

No action is required.

CKZ11342W USER CATALOG NOT IN CATALOG LIST - COMPONENT NAME=dsname | BCS=user catalog

Explanation:

A VVDS entry had a BCS backpointer which was not in the list of source user catalogs. Processing continues. The data set will be handled as a 'missing ucat' data set.

User response:

No action is required.

CKZ11343W UNABLE TO RENAME DSN=source datasetname | USING MASK=target mask

Explanation:

The new name of a data set will exceed 44 characters. Processing continues.

User response:

No action is required.

CKZ11344E VVDS ENTRY NOT FOUND FOR dsname

Explanation:

An expected VVDS entry was not found for the dsname. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11345E ERROR ACCESSING VVDS, LOC=lllll

Explanation:

An error occurred processing the VVDS on the volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11347E DSI CELL NOT FOUND IN VVDS RECORD FOR dsname

Explanation:

An expected VVDS cell was not found for the dsname. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11348E ERROR CALLING CKZ00902; R15=nnnn NEW DSN=target dsname

Explanation:

An error occurred calling CKZ00902 to obtain ACS information for the target data set name. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11349E VVDS WITH DSN=dataset NOT FOUND ON VOLUME=volser

Explanation:

The VVDS with name of data set was not found on volume volser. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11351I RENAMED CATALOG IS NOT USABLE: dsname

Explanation:

A catalog on a target volume was included in a RENAME-MASKS entry. The cloned catalog is not usable.

User response:

No action is required.

**CKZ11352E UPDATED VVDS ENTRY EXCEEDS
MAXIMUM ALLOWABLE LENGTH**

Explanation:

An updated VVDS entry was found to be larger than the maximum allowable size. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ11353I DUMP OF UPDATED VVDS ENTRY:

Explanation:

A dump of the VVDS entry follows.

User response:

No action is required.

**CKZ11360E ERROR DURING *function* FOR
VOLSER=*volume* - RETURN
CODE=*return_code* REASON
CODE=*reason_code***

Explanation:

An error occurred using IOSCAPU, UCBLOOK, or UCBPIN. Processing terminates.

User response:

A return code of 4 from UCBLOOK may indicate that the volume is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11367E *type* VOLUME *volume* IS NOT
ONLINE**

Explanation:

The indicated volume is not online. *type* indicates if this is a source or a target volume. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11401I hh:mm:ss VOLUME CLEANUP
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss
VOLUME CLEANUP COMPLETED;
RETURN CODE=nnn**

Explanation:

VOLUME CLEANUP processing message.

User response:

No action is required.

**CKZ11405E ALLOCATION FAILED FOR
VOLUME: *volume serial***

Explanation:

Dynamic allocation for a volume failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11405W DEALLOCATION FAILED FOR
DDNAME: *ddname***

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11406E ERROR CALLING CKZ01VV2
tablename FUNCTION: *function*
R15=*register15* R0=*register0*
LOC=*location***

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11407W ERROR CALLING CKZ01HEX;
FUNCTION: *function* R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ11411E JOURNAL FILE IS EMPTY

Explanation:

An attempt was made to read the journal file. The file is empty. Processing terminates.

User response:

Verify that the journal file has been correctly specified.

**CKZ11430W DELETE FAILED. RETURN
CODE=*return_code* REASON
CODE=*reason_code*
VOLSER=*target_volser*
DSN=*source_data_set_name*
ORIGINAL
DSN=*original_data_set_name***

Explanation:

An attempt was made to delete a data set from a target volume. The request to SSI failed. Processing continues.

User response:

The target volumes are usable. There are either unrenamed data sets or temporary data sets that were not deleted.

**CKZ11435I PROCESSING
VOLSER=targetvolser**

Explanation:

Volume cleanup informational message.

User response:

No action is required.

**CKZ11436I DELETE WILL BE ISSUED FOR
DSN=sourcedatasetname**

Explanation:

During a RENAME SIMULATE run, informational message issued for data sets which will be deleted from the target volumes during the actual RENAME run.

User response:

No action is required.

**CKZ11501I hh:mm:ss VOLUME RESTORES
STARTED - PROGRAM REV=rrr |
hh:mm:ss VOLUME RESTORES
COMPLETED; RETURN CODE=nnn**

Explanation:

VOLUME RESTORES processing message.

User response:

No action is required.

**CKZ11503I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ11504E OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11505E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11506E ERROR CALLING CKZ01VV2
tablename FUNCTION: function
R15=register15 R0=register0
LOC=location**

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ11508E UNABLE TO LOAD PROGRAM:
program_name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ11510E COMMAND FAILED: command IN
MODULE CKZ00047**

Explanation:

An error occurred while calling CKZ00047. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11523E ERROR OPENING DDNAME
DRSTATS**

Explanation:

An error occurred opening sysout file DRSTATS. Processing terminates.

User response:

Check that the appropriate DD statement is in the JCL. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11524E VOLUME RESTORE FAILED FOR
volser. SEE SYSOUT DRSTATS FOR
MORE INFORMATION**

Explanation:

An error occurred during RERUN mode attempting to restore a volume. Processing terminates.

User response:

Check the DRSTATS for the error that occurred. If unable to correct the problem, contact IBM Software Support.

**CKZ11525E VOLUME *volume* WAS ALREADY
RESTORED**

Explanation:

The volume that is listed in the message was already restored. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11526E VOLUME *volume* IS MARKED AS
NOT BACKED UP**

Explanation:

The volume that is listed in the message is marked as not backed up. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11528E VOLUME *volume* IS NOT FOUND IN
BACKUP**

Explanation:

A volume is marked as backed up in the JOURNAL, but corresponding records were not found in the BACKUP file.

User response:

Verify that the JOURNAL and BACKUP files are correct. If unable to correct the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11531W REVERTING TO USE VOLBKUP
DIRECTLY**

Explanation:

An error occurred using a subset of the VOLBKUP file. The VOLBKUP file will be used rather than a subset for volume metadata restores. Processing continues.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ11532E VOLUMES IN VOLBKUP NOT IN
EXPECTED SEQUENCE**

Explanation:

An error occurred using the VOLBKUP file. The volume data in the VOLBKUP file was not in the expected sequence. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ11533E UNEXPECTED END OF FILE ON
VOLBKUP**

Explanation:

An error occurred using the VOLBKUP file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ11560E ERROR DURING function FOR
VOLSER=*volume* - RETURN
CODE=*nnnn* REASON CODE=*nnnn***

Explanation:

An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ11601I hh:mm:ss CHECK TARGETS
STARTED - PROGRAM REV=*rrr* |
hh:mm:ss CHECK TARGETS
COMPLETED; RETURN CODE=*nnn***

Explanation:

CHECK TARGETS processing message.

User response:

No action is required.

**CKZ11607W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=*nnnn***

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ11609E ERROR ACCESSING JOURNAL
FILE; LOC=IIIII**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ11611E JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11660E UCBSCAN ERROR; RETURN
CODE=nn REASON CODE=nn |
UCBINFO ERROR; RETURN
CODE=nn REASON CODE=nn**

Explanation:

An error occurred using UCBSCAN or UCBINFO. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ11676E DEVICE TYPE AND MODEL NOT
FOUND FOR volser**

Explanation:

IOSCDs for the volser did not return information about the target volume.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11679E TARGET VOLSER volser WAS NOT
FOUND ONLINE**

Explanation:

The indicated target volume serial was not found online on the system running RENAME. Processing terminates.

User response:

Check that all the target volumes are online to the system which will be running RENAME.

**CKZ11681E VOLUME volser DEVICE NUMBER
nnnn HAS NO chpid PATHS
AVAILABLE**

Explanation:

The indicated device has no paths available for Db2 Cloning Tool to use to access the volume. Processing terminates.

User response:

Ensure that at least one chpid is online for each target device specified.

**CKZ11701I hh:mm:ss CATALOG VCLOSE
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss
CATALOG VCLOSE COMPLETED;
RETURN CODE=nnn**

Explanation:

CATALOG VCLOSE processing message.

User response:

No action is required.

**CKZ11703I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ11704E OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11705W ALLOCATION FAILED FOR
DDNAME: ddname |
DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic allocation for a data set failed, or, dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. If an allocation failure occurs, processing continues without using CATALOG VCLOSE services. If a deallocation failure occurs, processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software

Support. Have available the listing containing these messages.

**CKZ11707W ERROR CALLING CKZ01HEX;
FUNCTION= function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ11720I CONSOLE name ACQUIRED FOR
CATALOG VCLOSE COMMANDS**

Explanation:

A console session has been acquired so that RENAME can issue MODIFY CATALOG,VCLOSE commands.

User response:

No action is required.

CKZ11721I CONSOLE name FREED

Explanation:

The operator console has been freed.

User response:

No action is required.

CKZ11722W UNABLE TO ACQUIRE A CONSOLE

Explanation:

RENAME has failed to acquire a console for performing operator commands. Processing continues without using a console to perform operator commands.

User response:

No action is required.

**CKZ11723I CONSOLE name ALREADY IN USE,
WILL TRY ANOTHER**

Explanation:

The displayed name is already in use, probably from another copy of RENAME. RENAME will increment the number portion of the name and try again.

User response:

No action is required.

CKZ11724W MCS ALERT RECEIVED; text

Explanation:

An alert has been received for the console. Text describes the type of alert. Processing continues.

User response:

No action is required.

**CKZ11727I NON RESPONSE MDBS RECEIVED:
nnn**

Explanation:

Informational message that displays the number of received messages that were not a response to the command issued.

User response:

No action is required.

**CKZ11731W MCSOPER ERROR; FUNCTION=
function RC=nnnn RSN=nnnn |
MCSOPMSG ERROR; FUNCTION=
function RC=nnnn RSN=nnnn**

Explanation:

An error occurred using MCSOPER or MCSOPMSG. Processing continues.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ11740I COMMAND: text

Explanation:

Display operator command being issued.

User response:

No action is required.

**CKZ11741W WAIT TIME EXCEEDED FOR
COMMAND RESPONSE**

Explanation:

A response to the operator command was not received in a timely manner. Processing continues.

User response:

Determine if the catalog address space (CAS) is not responding to modify commands or is unable to process them in a timely manner. If unable to determine the cause, please report this message to IBM Software Support. Have available the listing containing this message.

**CKZ13001I hh:mm:ss SORTS STARTED -
PROGRAM REV=rrr | hh:mm:ss
SORTS COMPLETED; RETURN
CODE=nnn**

Explanation:

BCS backup sorts processing message.

User response:

No action is required.

**CKZ13003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ13004E DDNAME MISSING: ddname |
OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ13005E ALLOCATION FAILED FOR DSN:
 datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ13005W DEALLOCATION FAILED FOR
 DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ13006E ERROR CALLING CKZ01VV2
 tablename FUNCTION: function
 R15=register15 R0=register0
 LOC=location**

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ13007W ERROR CALLING CKZ01HEX;
 FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ13009E ERROR ACCESSING JOURNAL
 FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ13011E JOURNAL USER CATALOG
 RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ13012E UNABLE TO ESTABLISH ESTAEX;
 R15=nnnn LOC=lllll**

Explanation:

A subroutine was not able to establish an estaex environment. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ13030I SORT FOR SOURCE BCS RECORDS
 SUCCESSFUL; BCS=bcs dsname**

Explanation:

The sort of the indicated BCS's records was successful.

User response:

No action is required.

**CKZ13031E SORT FOR SOURCE BCS RECORDS
 FAILED; R15=nnnn; BCS=bcs
 dsname**

Explanation:

The sort of the indicated BCS's records failed. Processing terminates.

User response:

Correct the problem that caused SORT to fail. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ13041W NO BCS ENTRIES MATCH RENAME
 CRITERIA FOR SOURCE BCS=BCS
 dsname**

Explanation:

No BCS record keys matched the source dsname(s)/mask(s) specified. Processing continues.

User response:

Verify that the correct USERCATALOGS were specified in the COPY command and that the correct source names were specified in the RENAME-MASKS.

**CKZ13042W DUPLICATE KEYS IN BACKUP;
FIRST RECORD KEPT**

Explanation:

A record with a duplicate key was detected from the backup file. The records either do not contain a creation date or the creation dates are equal. The second record was dropped. Both records are printed. Processing continues.

User response:

Examine the printed records. The dropped record may have the desired data.

**CKZ13043E BCS BACKUPS DID NOT
COMPLETE; CHECK COPY STEP**

Explanation:

The journal indicates that the BCS backups did not complete. Processing terminates.

User response:

Check that the COPY command has completed successfully before initiating the RENAME command.

**CKZ13044E UNEXPECTED EOF ON BCS
BACKUP FILE; BCS=bcs dsname**

Explanation:

A logical 'end of data' condition is expected; a physical end of file was detected.

User response:

Check that the COPY command has completed successfully before initiating the RENAME command.

**CKZ13045E NO BCS ENTRIES MATCH RENAME
CRITERIA FOR ANY SOURCE BCS**

Explanation:

No BCS record keys matched the source dsname(s)/mask(s) specified. Processing terminates.

User response:

Verify that the correct USERCATALOGS were specified in the COPY command and that the correct source names were specified in the RENAME-MASKS.

CKZ13046E INTERNAL ERROR: *error_text*

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ13099E ABEND DURING BCS SORT

Explanation:

An abend occurred for a BCS SORT. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ13501I hh:mm:ss BCS CLEANUP STARTED
- PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss BCS
CLEANUP COMPLETED; RETURN
CODE=nnn**

Explanation:

BCS CLEANUP processing message.

User response:

No action is required.

**CKZ13503I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ13504E OPEN FAILED FOR
DDNAME=BCSRECS**

Explanation:

BCSRECS did not open successfully. Processing terminates.

User response:

Check that //BCSRECS points to the correct data set. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing containing this message.

**CKZ13505E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ13505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ13507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ13509E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ13511E JOURNAL CONTROL RECORD NOT
FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ13520W KEY OF ZEROS FOUND; ENTRY
BYPASSED**

Explanation:

An entry was found in BCSRECS with a key of zeros. The entry is bypassed. Processing continues.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ13521E KEY OF ZEROS RETURNED

Explanation:

The BCS entry with a key of zeros (binary) was returned; it was not requested. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ13522I RECORDS DELETED FROM BCS(S),
COUNT=nnn**

Explanation:

The indicated number of BCS records have been removed from the target BCS(s).

User response:

No action is required.

**CKZ13523I ENTRIES DELETED BY SVC26,
COUNT=nnn**

Explanation:

The indicated number of BCS entries have been removed from the target BCS(s) using SVC26.

User response:

No action is required.

**CKZ13544I THE TARGET CATALOG IS IN AN
INVALID STATE; THE CATALOG
WILL BE BYPASSED: catalog**

Explanation:

The catalog that is listed in the message is in an invalid state, and will be bypassed for cleanup.

User response:

No action is required.

**CKZ13545E ERROR ACCESSING BCS=bcs
dsname; LOC=lllll**

Explanation:

A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response:

See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ13549E DELETE FAILED; R15=nnn
REASON CODE=rrr MODULE=mm
DSN=dsname**

Explanation:

The SVC26 delete for a data set failed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ13599E ABEND DURING DB2 CLONING
TOOL PROCESSING**

Explanation:

An abend occurred during BCS cleanup. Processing terminates.

User response:

Determine the reason for the abend and correct if possible. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ13560I WAITING FOR EXCLUSIVE
CONTROL OF BCS bcs name**

Explanation:

The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response:

No action is required.

CKZ13561E **UNABLE TO ALLOCATE BCS: bcs name; WAIT TIME LIMIT EXCEEDED**

Explanation:

The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response:

Change the scheduling of the jobs so the Db2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the Db2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ13601I **Catalog orphan cleanup started - program rev=REV ** SIMULATION ** | Catalog orphan cleanup started - return code= *return_code*.**

Explanation

This message indicates one of the following:

- In SIMULATION mode, it indicates the start of the BCSCLEAN catalog orphan phase for keywords CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS. The string ** SIMULATION ** is appended to the message text.
- In a normal run, this message indicates the end of the BCSCLEAN catalog orphan phase with given return code.

User response:

No action is required.

CKZ13603I **DD name= *ddname* allocated for user catalog *user_catalog*.**

Explanation:

The user catalog listed in the message text has been dynamically allocated to the DD name listed in the message.

User response:

No action is required.

CKZ13604I **DD name= *ddname* allocated for volume *volume*.**

Explanation:

The volume specified in the messages text has been dynamically allocated to the DD name *ddname*.

User response:

No action is required.

CKZ13605E **Allocation failed for user catalog *user_catalog*.**

Explanation:

Dynamic allocation of the user catalog listed in the message failed. Processing stops.

User response:

Verify that the user catalog name is correct and accessible. Check for other error messages that may have been issued. If unable to determine the cause of this error, contact IBM Software Support.

CKZ13605W **Deallocation failed for DD name: *ddname*.**

Explanation:

Dynamic deallocation of the DD name listed in the message text failed.

User response:

Check for other error messages that may have been issued. If unable to determine the cause of this error, contact IBM Software Support.

CKZ13608E **Allocation failed for volume *volume*.**

Explanation:

Dynamic allocation of the volume listed in the message text failed. Processing stops.

User response:

Verify that the volume name is correct and accessible. Check for other error messages that may have been issued. If unable to determine the cause of this error, contact IBM Software Support.

CKZ13609E **Error accessing journal file; LOC= *internal_location*.**

Explanation:

An error occurred accessing the journal file. Processing stops.

User response:

Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ13610I **DD name= *ddname* deallocated.**

Explanation:

The DD name listed in the message text has been deallocated.

User response:

No action is required.

CKZ13611E **Journal *record_type* record *error_details*.**

Explanation:

The record type listed in the message produced an error that is described by the string *error_details*. Processing stops.

User response:

Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ13612E Journal volume pair record count does not match count found.**Explanation:**

The number of volume pair records found does not match the expected number from the control record. Processing stops.

User response:

Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ13615I Target user catalogs for this BCSCLEAN job: list.**Explanation:**

All user catalogs from the journal are listed and will be included in this execution.

User response:

No action is required.

CKZ13616I Target volumes for this BCSCLEAN job: list.**Explanation:**

All target volumes from the journal are listed and will be included in this execution.

User response:

No action is required.

CKZ13617E SVC99 error encountered for dsn: data_set_name – details follow:**Explanation:**

Dynamic allocation failed for the given data set name. A detailed set of dynamic allocation error messages (up to nine detail lines) follows this message. Processing stops.

User response:

Evaluate the error messages and try to resolve the error condition. If unable to resolve the issue, contact IBM Software Support. Have available the listing that contains these messages.

CKZ13618I Skipping type with dsn data_set_name: extension records not supported.**Explanation:**

The catalog record for the given type and data set name spans across multiple physical catalog records, which is not supported. Processing continues, but this data set is ignored.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ13619I No orphan records were found - no deletes are required.**Explanation:**

No orphan data sets were found that require action.

User response:

No action is required.

CKZ13620I Total orphan records checked: number_of_records.**Explanation:**

This message lists the total number of catalog records that were examined for volume or data set name mask comparison or both.

User response:

No action is required.

CKZ13621I Orphan records checked in user catalog: number_of_records.**Explanation:**

This message lists the number of catalog records within the current user catalog that were examined for volume or data set name mask comparison or both.

User response:

No action is required.

CKZ13622I Total orphan records deleted: number_of_records.**Explanation:**

This message lists the total number of catalog records that were deleted due to a match on target volume per the CLEANUP-CATALOG-ORPHANS keyword.

User response:

No action is required.

CKZ13623I Total orphan dsnmask entries deleted: number_of_records.**Explanation:**

This message lists the total number of files deleted that matched any CATALOG-CLEANUP-DSNMASK data set name mask value.

User response:

No action is required.

CKZ13624I Start of user catalog processing: user_catalog_name.**Explanation:**

The user catalog listed in the message is open for processing.

User response:

No action is required.

CKZ13625I End of user catalog processing: user_catalog_name.**Explanation:**

The user catalog listed in the message is closed and processing is complete.

User response:
No action is required.

CKZ13628I Start of delete processing .

Explanation:
This message indicates the start of the delete phase. The files identified by CLEANUP-CATALOG-ORPHANS (if specified) will be deleted with the NOSCRATCH option. The files identified by CLEANUP-CATALOG-DSNMASK will be deleted with the SCRATCH option if they reside on a non-target volume; they will be deleted with the NOSCRATCH option if they reside on a target volume.

User response:
No action is required.

CKZ13629I End of delete processing.

Explanation:
This message indicates the end of the delete phase.

User response:
No action is required.

**CKZ13631E AMSOPEN failed.
R15=return_code.**

Explanation:
The AMSOPEN macro failed with the given return code. Processing stops.

User response:
Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

**CKZ13632I Non-zero returned by IDCAMS;
RC=return_code.**

Explanation:
IDCAMs returned an unexpected non-zero return code, as shown in message. Processing stops.

User response:
Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

**CKZ13644I THE TARGET CATALOG IS IN AN
INVALID STATE; THE CATALOG
WILL BE BYPASSED: catalog**

Explanation:
The catalog that is listed in the message is in an invalid state, and will be bypassed for cleanup of catalog orphans.

User response:
No action is required.

**CKZ13645E Error accessing BCS=user_catalog
loc= internal_location.**

Explanation:

The user catalog (BCS) could not be read at the internal location listed in the message text.

User response:
Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

**CKZ13648I Delete: record_type
data_set_name sim
diagnostic_information.**

Explanation:
The file *data_set_name* with the catalog record type *record_type* was deleted. If '** SIMULATION **' appears in the message text, no delete is performed. Additional diagnostic information may be included in the message text (*diagnostic_information*).

User response:
No action is required.

**CKZ13660I Waiting for exclusive control of
BCS user_catalog.**

Explanation:
Contention was encountered when trying to allocate the user catalog. Processing waits for the interval specified in the value of INI keyword CONCURRENT_EXECUTIONS_WAIT_TIME.

User response:
No action is required.

**CKZ13661E Unable to allocate BCS
user_catalog; wait time limit
exceeded.**

Explanation:
Allocation failed for the user catalog due to contention and a subsequent exhaustion of a retry loop, as directed by the current value of INI keyword CONCURRENT_EXECUTIONS_WAIT_TIME.

User response:
Determine the source of the contention and the process that is holding the resource. Consider lengthening the time interval for the INI keyword CONCURRENT_EXECUTIONS_WAIT_TIME. If no resolution can be found, contact IBM Software Support. Have available the journal file and the listing that contains these messages.

**CKZ13663I Type: record_type DSN:
data_set_name match on vol:
volser.**

Explanation
This informational message can appear if the CLEANUP-CATALOG-ORPHANS keyword is specified for the BCSCLEAN command. It indicates that the target catalog entry that belongs to data set *data_set_name* on volume *volser* was not created by a

prior cloning, but the entry will be removed because the CLEANUP-CATALOG-ORPHANS keyword was specified. The *record_type* is the type of catalog entry and may have the following values:

- C** VSAM cluster
- B** GDG base structure
- A** Non-VSAM data set

User response:
No action is required.

CKZ13664I **Type: *record_type* Match dsn:
data_set_name Mask:
data_set_name_mask.**

Explanation

This informational message can appear if the CLEANUP-CATALOG-DSNMASKS keyword is specified for the BCSCLEAN command. It indicates that the target catalog entry that belongs to data set *data_set_name*, which is not on any target volume, will be removed because its data set name matches *data_set_name_mask* that was specified in CLEANUP-CATALOG-DSNMASKS. The *record_type* is the type of catalog record and may have the following values:

- C** VSAM cluster
- B** GDG base structure
- A** Non-VSAM data set

User response:
No action is required.

CKZ14001I **hh:mm:ss BCS UPDATES STARTED
- PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss BCS
UPDATES COMPLETED; RETURN
CODE=nnn**

Explanation:
BCS update processing message.

User response:
No action is required.

CKZ14006E **ERROR CALLING CKZ01VV2
tablename FUNCTION: function
R15=register15 R0=register0
LOC=location**

Explanation:
A problem occurred using a dataspace. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ14007W **ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:
An error occurred using CKZ01HEX to print a record. Processing continues.

User response:
Please report this message to IBM Software Support.

CKZ14008E **UNABLE TO LOAD PROGRAM:
program name | UNABLE TO LINK
TO PROGRAM: program name**

Explanation:
The indicated program name was not found. Processing terminates.

User response:
Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ14009E **ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:
A VSAM error occurred accessing the journal file. Processing terminates.

User response:
See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ14011E **JOURNAL CONTROL RECORD NOT
FOUND**

Explanation:
An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ14030I **BCS UPDATE COMPLETED;
RETURN CODE=nn SYSOUT
DD=sysout ddname FOR TARGET
BCS=bcs dsname | BCS UPDATE
FAILED ; RETURN CODE=nn
SYSOUT DD=sysout ddname FOR
TARGET BCS=bcs dsname**

Explanation:
The BCS update for the indicated target BCS has ended.

User response:

No action is required if the RETURN CODE is zero. If the RETURN CODE is not zero, check the indicated sysout file for warning or error messages related to the BCS update.

CKZ14031I DB2RECS UPDATED FOR TARGET BCS=

Explanation:

The DB2 Catalog and DB2 Directory data sets entries have been written to the DB2RECS file.

User response:

No action is required.

CKZ14032E NO DATA SETS MATCHED THE SPECIFIED DB2PREFIX: db2prefix

Explanation:

No data sets for any target usercatalog matched the high-level qualifier specified in DB2PREFIX. No entries were written to DB2RECS. Processing terminates.

User response:

Correct the name specified in the DB2PREFIX keyword.

CKZ14033W BCS UPDATE BYPASSED FOR TARGET BCS=bcs dsname ; error text SOURCE BCS=bcs dsname

Explanation:

The BCS update for the indicated target/source BCS has not been done. The error text indicates the reason.

User response:

No action is required.

CKZ14040E UNEXPECTED CONDITION; error text

Explanation:

An unexpected condition occurred while dispatching a BCS update task or while waiting for the completion of a BCS update task. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14041E SORT FOR BCS RECORDS DID NOT COMPLETE SUCCESSFULLY; BCS=bcs dsname

Explanation:

The sort of the records for the indicated BCS was not successful. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14089I NOT ALL DATASETS HAVE BEEN CATALOGED

Explanation:

An error occurred during BCS rename processing that caused some data sets to not be cataloged in a target catalog. Message CKZ14030I will indicate which catalog had an error.

User response:

Correct the cause of the error and rerun RENAME if possible.

CKZ14101I hh:mm:ss BCS UPDATE STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss BCS UPDATE COMPLETED; RETURN CODE=n RECORD COUNT=nnn**

Explanation:

BCS UPDATE task processing message.

User response:

No action is required.

CKZ14103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ14104E OPEN FAILED FOR DDNAME=ddname

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ14105E ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ14105W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ14106E ERROR CALLING CKZ01VV2
tablename FUNCTION: function
R15=register15 R0=register0
LOC=location**

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ14107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ14109E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ14110E DUPLICATE JOURNAL ENTRY;
LOC=lllll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14111E JOURNAL USER CATALOG
RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14112E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn**

Explanation:

The program was not able to establish an estaex environment. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14122I IDCAMS WILL BE USED FOR THIS
BCS**

Explanation:

Informational message. BCS extension records will cause program CKZ00150 to invoke IDCAMS.

User response:

No action is required.

**CKZ14123I EXPIRATION DATE IGNORED FOR
GDG ENTRY; GDG BASE
NAME=gdg base name**

Explanation:

Informational message. The expiration date of the GDG base has been ignored.

User response:

No action is required.

**CKZ14135I SMS smstypeCLAS COPIED FROM
SOURCE FOR BCS ENTRY - dsname**

Explanation:

No default value was given for the smstype (DATA, MGMT, or, STOR). The indicated SMS class for the entry was copied from the source data set.

User response:

No action is required, unless a specific class is desired.

**CKZ14140E UNEXPECTED CONDITION; error
text**

Explanation:

An unexpected condition occurred. 'error text' has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14141I BCS ENTRY NOT ON SOURCE
VOLUME SERIALS; BCS KEY=bcs
key name**

Explanation:

A BCS entry matched the rename masks, but, the entry is not cataloged on the source volume serials.

User response:

No action is required.

CKZ14142E **BCS ENTRY PARTIALLY ON SOURCE VOLUME SERIALS; BCS KEY DSN=bcs key name | GDG HAS GDS ENTRIES THAT ARE NOT ON SOURCE VOLUME SERIALS; GDG BASE NAME=gdg base name**

Explanation:

The indicated BCS entry is only partially on the source volume serials. Processing terminates.

User response:

Ensure that data sets (VSAM spheres, GDSs associated with a base GDG) are wholly contained on the source volume serials.

CKZ14142W **GDG HAS MIGRATED GDS ENTRIES; GDG BASE NAME=gdg base name | GDG HAS TAPE GDS ENTRIES; GDG BASE NAME=gdg base name**

Explanation:

The indicated BCS entry is only partially on the source volume serials. Processing continues, but the target GDSs which are migrated or on tape will not be accessible.

User response:

Ensure that data sets (VSAM spheres, GDSs associated with a base GDG) are wholly contained on the source volume serials.

CKZ14143E **UNABLE TO RENAME DSN=source datasetname USING MASK=target rename mask**

Explanation:

The new name of a data set will exceed 44 characters, or, the new name of a GDG base will exceed 35 characters. Processing terminates.

User response:

Correct the RENAME-MASKS specification.

CKZ14144E **NO SOURCE DSN/MASK MATCH FOUND FOR DSN=datasetname ENTRY IS PART OF BCS KEY=bcs key name**

Explanation:

A BCS entry did not fully match the specified rename masks. Processing terminates.

User response:

Ensure the RENAME-MASKS specification includes all spheres of VSAM components and all aliases for non-VSAM entries.

CKZ14145E **ERROR ACCESSING BCS=bcs dsname; LOC=lllll**

Explanation:

A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response:

See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ14146W **BCS ENTRY SKIPPED; RECORD TYPE NOT SUPPORTED**

Explanation:

Probably, a 'U' type BCS entry was detected. The entry is printed. Db2 Cloning Tool does not support renaming user catalogs on the target volume serials.

User response:

No action is required.

CKZ14147E **DUPLICATE BCS ENTRY | AN ENTRY FOR name ALREADY EXISTS IN THE TARGET BCS**

Explanation:

A duplicate record was detected when adding an entry to the target user catalog. The entry already exists in the target user catalog. The existing entry could be there from a prior run of Db2 Cloning Tool where BCSCLEAN was not used to remove the entry, or the RENAME-MASKS being used caused duplicate data set names to be created, or the existing entry could belong to a data set that was created prior to the Db2 Cloning Tool run and is not on a target volume. The duplicate catalog entry is printed. Processing terminates.

User response:

Ensure the RENAME-MASKS are not causing duplicate data set names to be created. If the RENAME-MASKS are not causing duplicate data set names, determine why the entry already exists in the target user catalog. To replace existing entries in the target user catalog, use the RECATALOG(Y)keyword in the RENAME command.

CKZ14148E **ERROR CALLING CKZ00902; R15=nnnn NEW DSN=new datasetname**

Explanation:

An error occurred invoking the ACS routines for the indicated new data set name. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14149E **FAILED TO POSITION IN BCSRECS**

Explanation:

An error occurred attempting to position to the correct entry to update in the BCSRECS file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14150W GDG IS EMPTY; GDG BASE NAME=gdg base name | ALL GDG'S ARE MIGRATED; GDG BASE NAME=gdg base name

Explanation:

An empty GDG was encountered and GDG-EMPTY(RETAIN,RC(4)) was specified or all GDGs are migrated and GDG-ALL-MIGRATED(RETAIN,RC(4)) was specified.

User response:

No action is required.

CKZ14151E UNEXPECTED END OF FILE ENCOUNTERED ON BCSRECS

Explanation:

An error occurred attempting to read the entry to update in the BCSRECS file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14160I WAITING FOR EXCLUSIVE CONTROL OF BCS bcs name

Explanation:

The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response:

No action is required.

CKZ14161E UNABLE TO ALLOCATE BCS: bcs name; WAIT TIME LIMIT EXCEEDED

Explanation:

The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response:

Change the scheduling of the jobs so the Db2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the Db2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ14199E ABEND DURING BCS UPDATE

Explanation:

An abend occurred for a BCS update task. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14201I hh:mm:ss BCSRECS FORMAT STARTED - PROGRAM REV=rrr | hh:mm:ss BCSRECS FORMAT COMPLETED; RETURN CODE=nnn

Explanation:

BCSRECS file processing message.

User response:

No action is required.

CKZ14204E OPEN FAILED FOR DDNAME=BCSRECS

Explanation:

BCSRECS did not open successfully. Processing terminates.

User response:

Check that //BCSRECS points to the correct data set. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ14240E ERROR WRITING TO BCSRECS; R15=nnnn

Explanation:

An error occurred formatting the BCSRECS file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14301I hh:mm:ss BCS CONDITIONING STARTED - PROGRAM REV=revision | hh:mm:ss BCS CONDITIONING COMPLETED; RETURN CODE=return_code

Explanation:

This message is a BCS CONDITIONING processing message.

User response:

No action is required.

CKZ14303I DDNAME=ddname ALLOCATED FOR DSN=data_set_name

Explanation:

ddname has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ14305E ALLOCATION FAILED FOR DSN: data_set_name

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ14306W DEALLOCATION FAILED FOR
DDNAME: *ddname***

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ14307W ERROR CALLING CKZ01HEX;
FUNCTION: *function*
R15=*register15***

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14310E ERROR ACCESSING CATALOG;
DDNAME: *ddname* R15=*register15*
ERROR=*reason_code*
FUNCTION=*function* LOC=*location***

Explanation:

An error occurred accessing the catalog with the listed *function* with reason code *reason_code*. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ14320I CONDITIONING TARGET
CATALOG: *catalog***

Explanation:

The condition of target catalog *catalog* is displayed.

User response:

No action is required.

CKZ14321I RECORDS DELETED: *number*

Explanation:

The catalog was cleaned up; the number of deleted records is listed in the message.

User response:

No action is required.

**CKZ14322I TARGET CATALOG HAS ALREADY
BEEN CONDITIONED**

Explanation:

The target catalog has already been conditioned.

User response:

No action is required.

**CKZ14330I DATA/INDEX COMPONENT NAME
MATCHES TARGET VALUE; *dsn* |
DATA/INDEX VOLSER MATCHES
TARGET VALUE; *dsn* | DATA/
MGMT/STORCLAS MATCHES
TARGET VALUE; *dsn* | DATA/INDEX
TRUE NAME MATCHES TARGET
VALUE; *dsn* | VVDS NAME
MATCHES TARGET VALUE; *dsn***

Explanation:

This message provides information about the matched data set name.

User response:

No action is required.

**CKZ14335I SMS *type* CLAS COPIED FROM
SOURCE FOR BCS ENTRY -
*data_set_name***

Explanation:

The SMS class that is listed in the message was copied from the source for the BCS entry *data_set_name*.

User response:

No action is required.

CKZ14339E INTERNAL ERROR: *error_text*

Explanation:

This is an internal error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14340E UNEXPECTED RECORD
RETURNED; *reason***

Explanation:

An unexpected record was returned after verifying new records in the BCS of the target ICF catalog. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14341E EXPECTED RECORD NOT FOUND;
*reason***

Explanation

An expected record is not found in the BCS of the target ICF catalog that is defined on a target volume. The reason can be one of the following.

KEY 00 RECORD NOT FOUND

Cluster entry with key of X'00's was not found.

DATA TRUE RECORD NOT FOUND

True entry for data component was not found.

INDEX TRUE RECORD NOT FOUND

True entry for an index component was not found.

VVDS RECORD NOT FOUND

Cluster entry for the VVDS of the volume the catalog is on was not found.

Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14342E INVALID DATA IN RECORD;
 LOC=*location***

Explanation:

Invalid data was found in a new record in the BCS of the target ICF catalog. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ14343E NEW RECORD SHOULD NOT EXIST

Explanation:

A new record that should not yet exist was found in the BCS. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14344E EXISTING TRUE RECORD NOT AS
 EXPECTED**

Explanation:

The new true record was added in the BCS, but verification failed because the record is not the same as expected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14345E ERROR ACCESSING
 BCS=*data_set_name* LOC=*location***

Explanation:

An error occurred when accessing the BCS. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14346I TARGET VVDS RECORD ALREADY
 EXISTS *data_set_name***

Explanation:

The target VVDS record already exists.

User response:

No action is required.

**CKZ14348E ERROR CALLING CKZ00902;
 R15=*register15* NEW
 DSN=*data_set_name***

Explanation:

An error occurred calling CKZ00902 when retrieving SMS classes from ACS. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14360I WAITING FOR EXCLUSIVE
 CONTROL OF BCS *data_set_name***

Explanation:

The program is waiting for exclusive control of the BCS.

User response:

No action is required.

**CKZ14361E UNABLE TO ALLOCATE BCS:
 data_set_name; WAIT TIME LIMIT
 EXCEEDED**

Explanation:

The wait time limit for BCS allocation was exceeded. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15001I hh:mm:ss IDCAMS PROCESS
 STARTED - PROGRAM REV=*rrr* (**
 SIMULATION **) | hh:mm:ss
 IDCAMS PROCESS COMPLETED;
 RETURN CODE=*nnn* ENTRIES
 RECATALOGED=*nnn***

Explanation:

BCS IDCAMS processing message.

User response:

No action is required.

**CKZ15003I DDNAME=*ddname* ALLOCATED
 FOR DSN=*datasetname***

Explanation:

'*ddname*' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ15005E ALLOCATION FAILED FOR DSN:
datasetname****Explanation:**

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ15005W DEALLOCATION FAILED FOR
DDNAME: ddname****Explanation:**

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ15007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn****Explanation:**

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ15009E ERROR ACCESSING JOURNAL
FILE; LOC=lllll****Explanation:**

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ15011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL IDC RECORD(S)
NOT FOUND****Explanation:**

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15030E RETURN CODE 8 SET FOR
ORPHANCATENTRY****Explanation:**

A return code of 8 was requested for the indicated condition. Processing terminates.

User response:

No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ15031E AMSOPEN FAILED; R15=nnnn**Explanation:**

An attempt was made to issue an IDCAMS command. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15032I NON-ZERO RETURNED BY
IDCAMS; RC=nnnn****Explanation:**

An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will continue if ORPHANCATENTRY return code is less than or equal to 4. Processing will terminate if ORPHANCATENTRY return code is greater than 4.

User response:

No action is required.

**CKZ15035E DEVICE TYPE NOT RECOGNIZED;
DEVICE=X'dddddddd'; SOURCE
DSN=source datasetname****Explanation:**

The device type could not be converted to one for use by IDCAMS RECATALOG. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15040E UNEXPECTED CONDITION; error
text****Explanation:**

An unexpected condition occurred. 'error text' has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15041E MORE THAN nnn IDCAMS
MESSAGES****Explanation:**

More than nnn messages were returned for an IDCAMS command. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15045E ERROR ACCESSING VVDS=vvds
dsname; LOC=lllll**

Explanation:

A VSAM error occurred accessing the indicated VVDS. Processing terminates.

User response:

See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ15046W ENTRY NOT FOUND IN VVDS;
DSN=dsname**

Explanation:

A VSAM component was not found in the VVDS. The entry cannot be recataloged. Processing will continue if ORPHANCATENTRY return code is less than or equal to 4. Processing will terminate if ORPHANCATENTRY return code is greater than 4.

User response:

No action is required.

**CKZ15047W ENTRY COULD NOT BE
RECATALOGED; DSN=dsname**

Explanation:

A data set was not found on the target volume serials and cannot be recataloged. Processing will continue if ORPHANCATENTRY return code is less than or equal to 4. Processing will terminate if ORPHANCATENTRY return code is greater than 4.

User response:

No action is required.

**CKZ15048W ENTRY SKIPPED DUE TO CLUSTER
FAILURE; TYPE=type
DSN=dsname**

Explanation:

A type AIX® or PATH entry could not be recataloged because the associated base cluster could not be recataloged.

User response:

No action is required.

**CKZ15049E DELETE FAILED; R15=nnn
REASON CODE=rrr MODULE=mm
DSN=dsname**

Explanation:

The SVC26 delete for a data set failed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ15101I hh:mm:ss UPDATE IAM
ASSOCIATIONS PROCESS
STARTED - PROGRAM REV=rrr (****

**SIMULATION **) | hh:mm:ss
UPDATE IAM ASSOCIATIONS
PROCESS COMPLETED; RETURN
CODE=nnn ENTRIES
UPDATED=nnn**

Explanation:

Update IAM associations processing message.

User response:

No action is required.

**CKZ15103I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

ddname has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ15104E macro FAILED FOR
DDNAME=ddname loc=lllll**

Explanation:

The OPEN or RDJFCB z/OS macro failed for the identified *ddname*. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ15105E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ15105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ15106E ERROR CALLING CKZ01VV2 tttttt
FUNCTION: function R15=nnnn
RO=nnnnnnnn LOC=llll

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing these messages and the CKZINI member.

CKZ15107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ15109E ERROR ACCESSING JOURNAL
FILE; LOC=llll

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ15111E JOURNAL UIAM RECORD(S) NOT
FOUND | JOURNAL UIAM RECORD
IS WRONG VERSION

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151131E AMSOPEN FAILED; R15=nnnn

Explanation:

An attempt was made to issue an IDCAMS command. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151132I NON-ZERO RETURNED BY
IDCAMS; RC=nnnn

Explanation:

An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will

continue if the return code is 4 or less. Processing will terminate if the return code is 8 or above.

User response:

For return code 4 or less, none. For return code 8 or above see displayed IDC messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ151134E UNEXPECTED DATA FOUND IN
IAMPRINT reason

Explanation:

The parsing of the data in the IAMPRINT dd failed. The data in the IAMPRINT dd is different than expected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151141E MORE THAN nnn IDCAMS
MESSAGES

Explanation:

More than nnn messages were returned for an IDCAMS command. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151145E ASSOCIATED type FOR PATH
DSN=datasetname NOT PRESENT
type SOURCE DSN=datasetname

Explanation:

The AIX or CLUSTER data set associated with the identified PATH does not exist. It is possible to get this error when using RERUN when the rename mask entries that cover IAM data sets have changed. Processing terminates.

User response:

Verify the AIX or CLUSTER data set is on a cloned volume and is being renamed. If RERUN is being used and the rename mask entries that cover IAM data sets have changed, it may be necessary to start the cloning over from the beginning. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ15146E ASSOCIATED CLUSTER FOR AIX
DSN=datasetname NOT PRESENT
CLUSTER SOURCE
DSN=datasetname

Explanation:

The CLUSTER data set associated with the identified AIX does not exist. It is possible to get this error when using RERUN when the rename mask entries that cover IAM data sets have changed. Processing terminates.

User response:

Verify the CLUSTER data set is on a cloned volume and is being renamed. If RERUN is being used and the rename mask entries that cover IAM data sets have changed, it may be necessary to start the cloning over from the beginning. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ1S000I Program executed successfully

Explanation:

The stored procedure successfully completed.

User response:

No action is required.

CKZ1S001E Invalid type specified: *type*. Must be BUILDJCL, BUILD, CLEAN, CLONE, RECLONE or REMOVE

Explanation:

The stored procedure has been invoked with an unknown TYPE= request. Processing terminates.

User response:

Verify that a correct TYPE= request is being made to the stored procedure.

CKZ1S010E Open failed for DSN: *dsname(member)*, reason: *reason*

Explanation:

The data set name listed in the message cannot be opened. *reason* shows the failure reason returned by open.

User response:

Verify that the data set has the correct attributes. If unable to determine the reason that the data set open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S011E Close failed for DSN: *dsname(member)*, reason: *reason*

Explanation:

The data set name listed in the message cannot be closed. *reason* shows the failure reason returned by the close.

User response:

If unable to determine the reason that the data set close failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S012E Status file is empty: *dsname*

Explanation:

During an attempt to read the status file, the file was found to be empty.

User response:

Verify the correct status file data set is being used, the correct request type has been used, and the status file

has not been altered. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S013E Open failed for JCL: *dsname(member)*, reason: *reason*

Explanation:

The data set name that is listed in the message cannot be opened. *reason* shows the failure reason returned by the open.

User response:

Verify that the data set has the correct attributes. If unable to determine the reason that the data set open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S014E Close failed for JCL: *dsname(member)*, reason: *reason*

Explanation:

The data set name that is listed in the message cannot be closed. *reason* shows the failure reason returned by the close.

User response:

If unable to determine the reason that the data set close failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S015E Error reading from file: *dsname(member)*, reason: *reason*

Explanation:

The data set name that is listed in the message cannot be read. *reason* shows the failure reason that was returned by the read.

User response:

If unable to determine the reason that the data set read failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S020E Keyword from *dsname(member)*: Required parameter not found

Explanation:

The identified keyword is required, but is missing from the indicated parameter data set.

User response:

Add the missing keyword to the indicated parameter data set.

CKZ1S021E Keyword from *dsname(member)*: Required value was not set

Explanation:

The identified keyword in the indicated parameter data set did not have a value specified.

User response:

Correct the keyword in the indicated parameter data set to specify a value.

CKZ1S022E **Keyword from *dsname(member)*:
Value longer than *n* characters**

Explanation:

The identified keyword in the indicated parameter data set has a value that is too long. *n* shows the maximum length allowed for a value for this keyword.

User response:

Correct the keyword in the indicated parameter data set to specify a value that is not more than the maximum length.

CKZ1S023E **Keyword from *dsname(member)*:
Invalid value; Must be
allowed_values**

Explanation:

The identified keyword in the indicated parameter data set has a value that is not allowed. *allowed_values* shows the valid values that can be used.

User response:

Correct the keyword in the indicated parameter data set to specify a value that is allowed.

CKZ1S024E **Keyword from *dsname(member)*:
Invalid value; explanation**

Explanation:

The identified keyword in the indicated parameter data set has a value that is not allowed. *explanation* shows the allowable values that can be used.

User response:

Correct the keyword in the indicated parameter data set to specify a value that is allowed.

CKZ1S025E ***dsname(member)*: Keywords are
mutually exclusive: keyword1 and
keyword2**

Explanation:

The identified keywords in the indicated parameter data set have both been specified, but they are mutually exclusive. Only one of the keywords can be specified.

User response:

Remove one of the keywords from the parameter file.

CKZ1S026E **Keyword from *dsname(member)*:
Value has non-numeric characters**

Explanation:

The identified keyword in the indicated parameter data set has a value that is not numeric.

User response:

Correct the keyword in the indicated parameter data set to specify a numeric value.

CKZ1S027E **Keyword from *dsname(member)*:
Number of entries not a multiple of
*n***

Explanation:

The identified keyword in the indicated parameter data set has a value that contains entries that are not a correct multiple. *n* shows the expected multiple of entries.

User response:

Correct the keyword in the indicated parameter data set to specify a value that has the correct multiple of entries.

CKZ1S028E **Keyword from *dsname(member)*:
Value has more than 2 operands;
max of 2 allowed**

Explanation:

The identified keyword in the indicated parameter data set has a value that contains more than two entries. Only two entries are allowed.

User response:

Correct the keyword in the indicated parameter data set to specify a value that has no more than two entries.

CKZ1S029E **Keyword from *dsname(member)*:
Invalid value; keyword1 must be
only operand**

Explanation:

The identified keyword in the indicated parameter data set has a value that contains more than one entry. Only one entry is allowed.

User response:

Correct the keyword in the indicated parameter data set to specify a value that has no more than one entry.

CKZ1S030E **Keyword from *dsname(member)*:
Invalid value; Must begin with
allowed_values**

Explanation:

The identified keyword in the indicated parameter data set has a beginning value that is not allowed. *allowed_values* shows the allowable values that can be used.

User response:

Correct the keyword in the indicated parameter data set to specify a beginning value that is allowed.

CKZ1S031E **Keyword from *dsname(member)*:
Invalid value; Second must be
allowed_values**

Explanation:

The identified keyword in the indicated parameter data set has a second value that is not allowed. *allowed_values* shows the allowable values that can be used.

User response:

Correct the keyword in the indicated parameter data set to specify a second value that is allowed.

CKZ1S032E *dsname(member): keyword1 cannot be used with keyword2*

Explanation:

The identified keywords in the indicated parameter data set have both been specified, but they are mutually exclusive. Only one of the keywords can be specified.

User response:

Correct the keywords in the indicated parameter data set so that only one of the keywords is specified.

CKZ1S033E *dsname(member): Values required: list_of_values*

Explanation:

The identified keyword in the indicated parameter data set has a valid value. *list_of_values* shows the allowable values that can be used.

User response:

Correct the keyword in the indicated parameter data set to specify a valid value.

CKZ1S034E *dsname(member): Unknown keyword keyword*

Explanation:

The identified keyword in the indicated parameter data set is not a valid keyword.

User response:

Verify that the keyword is valid for the specific parameter data set and has not been misspelled.

CKZ1S035E *Keyword from dsname(member): Value is not allowed to contain embedded blanks*

Explanation:

The identified keyword in the indicated parameter data set has a value that contains blanks. Only a single value without embedded blanks is allowed.

User response:

Correct the keyword in the indicated parameter data set to specify a single value that does not contain embedded blanks.

CKZ1S040E *Admin task scheduler task already exists, TASK_NAME: task_name*

Explanation:

The addition of a task to the DB2 administrative task scheduler failed because the task name of the task being added is already in use. The task name may belong to another cloning, or a REMOVE was not performed to remove the tasks before a BUILD was done.

User response:

Verify that the task name is not in use by another cloning. If the task name should not be in use, manually delete the tasks from the administrative task scheduler.

CKZ1S041E *SYSPROC.ADMIN_TASK_ADD() procedure error. Return code: return_code, message: message*

Explanation:

An error has been returned from a call to the ADMIN_TASK_ADD stored procedure. Any messages that are returned by the ADMIN_TASK_ADD stored procedure are displayed.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S042E *SYSPROC.ADMIN_TASK_REMOVE() procedure error. Return code: return_code, message: message*

Explanation:

An error has been returned from a call to the ADMIN_TASK_REMOVE stored procedure. Any messages that are returned by the ADMIN_TASK_REMOVE stored procedure are displayed.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S043E *SYSPROC.ADMIN_TASK_UPDATE() procedure error. Return code: return_code, message: message*

Explanation:

An error was returned from a call to the ADMIN_TASK_UPDATE stored procedure. Any messages that were returned by the ADMIN_TASK_UPDATE stored procedure are displayed.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S050E *PassTicket generation error. Return code: return_code, message: message*

Explanation:

The generation of a PassTicket failed. The message describes the specific error from the PassTicket generation module.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S060E **The cloning has ended with error:**
TASK_NAME: *task_name*.
Message: *message*

Explanation:

A cloning job completed with an abend or an unexpected return code. The cloning stops. The message contains additional information about the error and the JES job ID of the cloning job.

User response:

To identify the problem, use the JES job ID to look at the sysout of the failed cloning job. If the JCL needs to be regenerated, the cloning must be cleaned up (CLEAN) and removed (REMOVE); then update the parameter files, BUILD, and CLONE. If the JCL does not need to be regenerated, the cloning must be cleaned up (CLEAN), and then rerun (CLONE or RECLONE).

CKZ1S070E **SQL error, SQLCODE:** *sql_code*

Explanation:

An error has been returned from an SQL call.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ1S101I **CLONING-TYPE = OFFLINE was specified for DB2SLB source; using CLONING-TYPE = ONLINE instead**

Explanation:

CLONING-TYPE = OFFLINE was specified with SOURCE-VOLUMES = DB2SLB. A DB2 system-level backup is taken while the source DB2 is active. Therefore, the cloning type must be ONLINE. A cloning type of ONLINE will be used.

User response:

No action is required.

CKZ20908E **LOAD failed for program: *program***
RC=X'*return_code*'
RSN=X'*reason_code*'

Explanation:

A LOAD macro failed to load the indicated program. *return_code* and *reason_code* show the return code and reason code from the LOAD macro invocation.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20910E **Must execute as APF authorized**

Explanation:

The stored procedure is not running in an APF authorized environment. The generation of PassTickets requires the stored procedure to run in an APF-authorized environment.

User response:

Verify that the STEPLIB of the WLM address space in which the stored procedure is running has only APF-authorized libraries.

CKZ20911E **Call type: *call_type*; Token in parm is invalid**

Explanation:

The parameter list used to call program CKZ00209 is invalid.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ20921E **TCB does not have an ACEE**

Explanation:

The task (TCB) of the stored procedure does not have an ACEE. The stored procedure task (TCB) is expected to have an ACEE.

User response:

Verify that the stored procedure was created with the SECURITY USER parameter. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20922E **ACEE has invalid user ID: *user_ID***

Explanation:

The user ID in the ACEE of the stored procedure task is less than one character or greater than eight characters. The user ID of the stored procedure task is expected to be 1-8 characters.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20923E **SET failed; SQLCODE: *sql_code***
SQLSTATE: *sql_state* LOC: *location*

Explanation:

A SQL SET statement has failed with the indicated SQLCODE and SQLSTATE.

User response:

If the SQLCODE is -805, verify that the bind has been run for CKZ00209. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20924E **DB2 administrative task scheduler is not defined in ZPARM**

Explanation:

The ZPARM for the DB2 system does not have the DB2 administrative task scheduler defined. The DB2 administrative task scheduler is required to run the stored procedure.

User response:

Verify that the DB2 administrative task scheduler is defined and configured on the DB2 system. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20925E **Not authorized to use user ID:**
user_id

Explanation:

The user ID that is running the stored procedure is not authorized to use the value of the USERID parameter that is specified in the cloning parameter file.

User response:

Change the value of the USERID in the cloning parameter file to an authorized value.

CKZ20926E **Unable to determine correct user ID to use; ACEE user ID:** *user_id*
Session user ID: *user_id*

Explanation:

The user ID in the ACEE of the stored procedure task is different from the session user ID. The user ID in the ACEE of the stored procedure task is expected to be the same as the session user ID.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20927E **PassTicket generation failed;**
RC=X'return_code'

Explanation:

The security server PassTicket generation routine returned an error with the return code that is listed in the message text.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20929E **Internal error; LOC:** *location*
reason

Explanation:

An internal error occurred. *location* is the location where the error occurred and *reason* is a brief description of the error.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ21501I **hh:mm:ss DB2GETBACKINFO**
STARTED - PROGRAM REV=rrr |
hh:mm:ss DB2GETBACKINFO
COMPLETED; RETURN CODE=nnn

Explanation:

DB2GETBACKINFO processing message.

User response:

No action is required.

CKZ21504E **DDNAME MISSING: ddname |**
OPEN FAILED FOR DDNAME:
ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ21505E **ALLOCATION FAILED FOR DSN:**
datasetname | DEALLOCATION
FAILED FOR DSN: datasetname

Explanation:

Dynamic allocation for a data set failed, or, dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ21507W **ERROR CALLING CKZ01HEX;**
FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ21508E **UNABLE TO LOAD PROGRAM:**
program name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ21509E **ERROR ACCESSING BSDS FILE;**
LOC=lllll

Explanation:

A VSAM error occurred accessing the BSDS file.
Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ21523I BACKUP DUMPTAPES FOR
LOCATION: *location* | BACKUP
DUMPTAPES FOR COPYPOOLS:
*copypools***

Explanation:

A list of SLB dump tapes follows.

User response:

No action is required.

CKZ21530I nnnn type RECORDS WRITTEN

Explanation:

Displays the number of type records written to the backinfo data set.

User response:

No action is required.

**CKZ21531E NO BACKUPS FOUND FOR
LOCATION: *location* [AND
DUMPCCLASS *dump_class*] | NO
BACKUPS FOUND FOR
COPYPOOLS: *copypools* [AND
DUMPCCLASS *dump_class*] |
REQUESTED BACKUP NOT USABLE
– NO DUMPTAPES FOUND**

Explanation:

No DB2 BACKUP SYSTEM backups were found in HSM for the indicated location. If the USE-DUMPTAPES keyword was supplied with DUMP-CLASS, and that dump class was not found, the requested dump class is listed in the message. For USE-DUMPTAPES, this message might be generated when the backup has an HSM DUMPSTATE that is not ALLCOMPLETE, or at least one dump tape for the backup has an expiration date prior to the current date; those backups are excluded from consideration.

User response:

Verify that the correct location or copypool names are being used. For USE-DUMPTAPES, ensure that the desired backup has dump tapes associated with it, that none of the tapes have expired, and that the dumpstate is ALLCOMPLETE. For USE-DUMPTAPES and DUMP-CLASS, ensure that the DUMP-CLASS keyword is correctly specified and that a usable backup exists for that DUMP-CLASS.

**CKZ21532E NO BACKUPS FOUND FOR TOKEN:
token**

Explanation:

No DB2 BACKUP SYSTEM backups were found in HSM for the indicated token.

User response:

Verify the correct token is being used.

**CKZ21533I BACKUPS FOR LOCATION: *location*
list of backups**

Explanation:

Displays the DB2 BACKUP SYSTEM backups that were found in HSM for the indicated location.

User response:

No action is required.

**CKZ21534W UNEXPECTED DATA FOUND IN
HSM RESPONSE**

Explanation:

The data found in the HSM response was not as expected.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ21535I SELECTED BACKUP

Explanation:

Displays information about the selected backup.

User response:

No action is required.

**CKZ21536E Some backup volumes are not on
DASD; VOLSER: *****.**

Explanation:

Some of the backup volumes for the selected backup have a volser of *****. A volser of ***** indicates that the backup volume is not on DASD. The selected backup cannot be used if any backup volume is not on DASD. Processing terminates.

User response:

Select a backup that has all the backup volumes on DASD.

**CKZ21537E BSDS RECORD NOT FOUND FOR
record-type**

Explanation:

A record of the indicated type was not found in the BSDS.

User response:

Verify the BSDS being used is valid. The LOCATION keyword can be used instead of BSDS-DDN. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages and a copy of the BSDS being used.

**CKZ21538I BSDS LOCATION: *location* WILL BE
USED**

Explanation:

Informational message that displays the location obtained from the BSDS.

User response:
No action is required.

**CKZ21539E LOCATE ERROR; RETURN
 CODE=nnnn X'hhhh' REASON
 CODE=nnnn X'hhhh'**

Explanation:
A LOCATE request failed. The return code and reason code are displayed in decimal and hexadecimal format. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ21540E CKZ00900 UNEXPECTED
 RESULTS; error text**

Explanation:
An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ21541E ARCHSEND ERROR; RETURN
 CODE=nnnn X'hhhh' REASON
 CODE=nnnn X'hhhh'**

Explanation:
An ARCHSEND request failed. The return code and reason code are displayed in decimal and hexadecimal format. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ21543E MULTIPLE DUMP CLASSES FOUND
 FOR SELECTED VERSION:
 dump_class1 AND *dump_class2* –
 DUMP-CLASS KEYWORD
 REQUIRED**

Explanation:
The DUMP-CLASS keyword was not specified, but because multiple dump classes were found, the DUMP-CLASS keyword is required.

User response:
Provide the DUMP-CLASS keyword, or select a different backup version that only has one dump class by using the TOKEN or LAST keywords.

**CKZ21545I USERCATALOG: user-catalog WAS
 FOUND ON BACKUP VOLUME:
 vvvvvv**

Explanation:

The indicated user catalog was found on the indicated backup volume.

User response:
No action is required.

**CKZ21546E USERCATALOG: user-catalog WAS
 NOT FOUND ON ANY BACKUP
 VOLUME**

Explanation:
The indicated user catalog was not found on any backup volume. The indicated user catalog may have been created after the backup that created the backup volumes. Processing terminates.

User response:
Verify the correct user catalogs have been specified in the USERCATALOGS keyword.

**CKZ21547E UNABLE TO DETERMINE VOLSER
 FOR USERCATALOG: user-catalog**

Explanation:
It is not possible to determine the current volser of the indicated user catalog. The user catalog may not exist. Processing terminates.

User response:
Verify the user catalog exists. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ21548E USERCATALOG: user-catalog IS
 NOT ON A COPIED VOLUME:
 vvvvvv**

Explanation:
The indicated user catalog was found to be currently on a volume that is not one of the source volumes. Processing terminates.

User response:
Verify the correct user catalogs have been specified in the USERCATALOGS keyword.

**CKZ21549I USERCATALOG: user-catalog WAS
 NOT FOUND ON EXPECTED
 BACKUP VOLUME: vvvvvv |
 USERCATALOG MAY HAVE MOVED
 SINCE BACKUP THAT CREATED
 BACKUP VOLUMES | SCANNING
 BACKUP VOLUMES TO FIND
 USERCATALOG**

Explanation:
The indicated user catalog was not found on the expected backup volume. The user catalog may have moved since the backup was run that created the backup volumes. The backup volumes will be scanned to find the user catalog. This process may take several minutes depending on the number of backup volumes that need to be scanned.

User response:
No action is required.

**CKZ21551E REQUIRED KEYWORD MISSING:
keyword**

Explanation:
A keyword required for processing has been omitted.
Processing terminates.

User response:
Specify the required keyword.

**CKZ21552E THE SAME *item* HAS BEEN
SPECIFIED FOR MULTIPLE
KEYWORDS: *value***

Explanation:
The indicated item has been specified in multiple
keywords. The specified values must all be different.
Processing terminates.

User response:
Specify different values in the keywords.

**CKZ21553E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:
The operand entered for a keyword exceeded the
maximum length allowed for the operand. nnn is the
maximum allowed length for the keyword. Processing
terminates.

User response:
Correct the length of the keyword's operand.

**CKZ21554E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:
Multiple operands were detected for a keyword; only
one operand is permitted. Processing terminates.

User response:
Correct the keyword to use one operand.

**CKZ21555E DUMP-CLASS KEYWORD
REQUIRES USE-DUMPTAPES
KEYWORD**

Explanation:
The DUMP-CLASS keyword was specified, but the USE-
DUMPTAPES keywords was not specified. USE-
DUMPTAPES is required with the DUMP-CLASS
keyword.

User response:
Provide the USE-DUMPTAPES keyword and resubmit.

**CKZ21556E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate
operand. Processing terminates.

User response:
Specify an appropriate operand for the keyword.

**CKZ21557E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation:
The indicated entry for the keyword was previously
specified. Processing terminates.

User response:
Remove the duplicate entry.

**CKZ21558E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:
The value in the keyword is invalid. 'error text'
indicates the problem detected with the value.
Processing terminates.

User response:
Correct the value specified in the keyword.

**CKZ21560E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn
LOC=lllll**

Explanation:
An error occurred during UCBLOOK processing.
Processing terminates.

User response:
Contact IBM Software Support. Have available the
listing containing this message.

CKZ21561E Internal error; Loc: *location reason*

Explanation:
An internal error has occurred. Processing terminates.

User response:
Contact IBM Software Support. Have available the
listing that contains this message.

**CKZ21569E WORK-DSN: dataset-name DOES
NOT EXIST OR IS NOT AVAILABLE
FOR USE**

Explanation:
The dynamic allocation of the indicated data set failed.
The data set may not exist or may be currently
allocated to another job. Processing terminates.

User response:
Verify the data set exists and is not currently allocated
to another job.

**CKZ21581I DSNS FOR KEYWORD: keyword list
of dsns**

Explanation:
Parsing found the listed dsns for the keyword.

User response:

No action is required.

CKZ21586I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword in the command.

User response:

No action is required.

**CKZ22001I hh:mm:ss DB2UPDATE STARTED -
PROGRAM REV=rrr | hh:mm:ss
DB2UPDATE COMPLETED; RETURN
CODE=nnn**

Explanation:

DB2UPDATE command processing message.

User response:

No action is required.

**CKZ22003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ22004E DDNAME missing: ddname | Open
failed for DDNAME: ddname**

Explanation:

ddname was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the DDNAME specified, or, add the appropriate DDNAME to the job's JCL.

**CKZ22005E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ22005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ22006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ22007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ22008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ22009E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ22010E DUPLICATE JOURNAL ENTRY;
LOC=lllll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL DSN MASK RECORD(S) NOT FOUND | JOURNAL VOLUME PAIR RECORD(S) NOT FOUND | JOURNAL DB2 CONTROL RECORD NOT FOUND | JOURNAL DB2 HLQ RECORD(S) NOT FOUND | JOURNAL DB2 STOGROUP RECORD(s) NOT FOUND

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

CKZ22012E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL DB2 CONTROL RECORD IS WRONG VERSION | JOURNAL VOLP RECORD IS WRONG VERSION | JOURNAL XMSK RECORD IS WRONG VERSION

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22013E RECORD COUNT IS ZERO; LOC=lllll | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc LOC=lllll

Explanation:

There was a problem with the journal records needed to initiate the DB2 update. For the first format, the journal control record indicate no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22014I COPY-BY-DS JOURNAL FOUND

Explanation:

The journal data set was created by COPY-BY-DS. Processing continues.

User response:

No action is required.

CKZ22015E THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY

Explanation:

The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response:

No action is required.

CKZ22016E DB2-HLQS ENTRIES MISMATCH; PRIMARY MEMBER HAD DIFFERENT DB2-HLQS ENTRIES

Explanation:

The entries entered for this member in keyword DB2-HLQS does not match the entries specified for the primary member's run of DB2UPDATE. The entries for the primary member's run are printed. Processing terminates.

User response:

Correct the entries to match those specified for the primary member.

CKZ22017E STOGROUPS ENTRIES MISMATCH; PRIMARY MEMBER HAD DIFFERENT STOGROUPS ENTRIES

Explanation:

The entries entered for this member in keyword STOGROUPS does not match the entries specified for the primary member's run of DB2UPDATE. The entries for the primary member's run are printed. Processing terminates.

User response:

Correct the entries to match those specified for the primary member.

CKZ22030I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how DB2UPDATE will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ22031E PRIOR COPY WAS A SIMULATION | PRIOR RENAME WAS A SIMULATION | PRIOR RENAME WAS NOT RUN

Explanation:

The journal indicates that the COPY command or the RENAME command was a simulation, or the RENAME command has not been run. Processing terminates.

User response:

Run DB2UPDATE after both the COPY and RENAME have successfully run in non-simulation.

CKZ22032E INTERNAL ERROR; FIELD TYPE NOT SET

Explanation:

The setting for the field type to be used was not recognized. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22033E SVC 99 INFORMATION RETRIEVAL FAILED; DDNAME: ddname R15=nnnn S99INFO= X'nnnn' S99ERROR= X'nnnn'

Explanation:

The SVC 99 information retrieval of the data set name of the identified DD has failed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22034E THE DATA SET NAME OF DDNAME: DBD01 DOES NOT HAVE DB2 FORMAT; datasetname

Explanation:

The data set name of the data set allocated to DD DBD01 does not have a DB2 data set name form. The directory information being updated by DB2UPDATE has been determined to no longer reside in the DBD01 table space. DB2UPDATE is attempting to dynamically allocate the SYSDBDXA table space where this data should reside but cannot determine what the data set name should be. Processing terminates.

User response:

Add a SYSDBDXA DD that points to the SYSDBDXA table space or point the DBD01 DD to a DBD01 table space that has a DB2 format data set name.

CKZ22035I NO DBD INFORMATION WAS FOUND IN DBD01; WILL DYNAMICALLY ALLOCATE SYSDBDXA TO USE

Explanation:

No DBD information was found in DBD01. The directory information being updated by DB2UPDATE has been determined to no longer reside in the DBD01 table space. DB2UPDATE will attempt to dynamically allocate the SYSDBDXA table space, where this data should reside, to use.

User response:

No action is required. A SYSDBDXA DD that points to the SYSDBDXA table space can be added to the JCL.

CKZ22041E CKZ00900 UNEXPECTED RESULTS; error text LOC=lllll

Explanation:

An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22042E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token error text for invalid value

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

CKZ22050E ERROR IN PARAMETERS FOR keyword

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ22051E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ22053E MAXIMUM LENGTH EXCEEDED FOR KEYWORD: keyword

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ22054E **KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword | KEYWORD HAS MORE THAN 2 OPERANDS; ONLY 2 ALLOWED: keyword | KEYWORD HAS MORE THAN 8 OPERANDS; ONLY 8 ALLOWED: keyword | KEYWORD HAS MORE THAN 40 OPERANDS; ONLY 40 ALLOWED: keyword**

Explanation:

More operands were specified for a keyword than are permitted. Processing terminates.

User response:

Correct the keyword.

CKZ22056E **NOTHING SPECIFIED FOR KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ22057E **DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ22058E **INVALID VALUE IN KEYWORD: keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ22060E **TARGET STORAGE GROUP NAME MUST BE THE SAME LENGTH AS THE SOURCE STORAGE GROUP NAME | TARGET STORAGE GROUP NAME: target-sgname | SOURCE STORAGE GROUP NAME: source-sgname**

Explanation:

The target storage group name is not the same length as the source storage group name. For DB2 Version 8, the names must be the same length. Processing terminates.

User response:

Correct the values specified so they are the same length.

CKZ22068E **UNPAIRED ENTRY IN KEYWORD: keyword**

Explanation:

For DB2-HLQS, there must be a source alias followed by a target alias. For DB2-MEMBERS, there must be a source member name followed by a target member name. For STOGROUPS, there must a source name followed by a target name. An uneven number of entries was specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ22069E **THE USE OF keyword1 REQUIRES THE USE OF keyword2**

Explanation:

Keyword1 was specified. Its use requires that keyword2 must also be specified. Processing terminates. An uneven number of entries was specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ22074E **NAME USED AS BOTH SOURCE AND TARGET: name IN KEYWORD: keyword**

Explanation:

A target entry was also specified as a source in the indicated keyword. Processing terminates.

User response:

Correct the keyword specification.

CKZ22075E **NAME USED AS TARGET MULTIPLE TIMES: name IN KEYWORD: keyword**

Explanation:

A target entry was specified for multiple sources in the indicated keyword. Processing terminates.

User response:

Correct the keyword specification.

CKZ22085I **PAIRS FOR KEYWORD: keyword**

Explanation:

The names indicated for the keyword have been accepted for processing.

User response:

No action is required.

CKZ22086I **VALIDATING KEYWORD: keyword**

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

CKZ22101I **hh:mm:ss BSDS UPDATING
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss BSDS
UPDATING COMPLETED; RETURN
CODE=nnn**

Explanation:

DB2UPDATE processing message.

User response:

No action is required.

CKZ22106E **ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ22107W **ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ22109E **ERROR ACCESSING BSDS FILE;
LOC=llll | ERROR ACCESSING
JOURNAL; LOC=llll**

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ22110E **DUPLICATE JOURNAL ENTRY;
LOC=llll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22111E **BSDS DATA SHARING RECORD(S)
NOT FOUND**

Explanation:

An expected record was not found in the BSDS file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22112E **JOURNAL DB2 CONTROL RECORD
NOT FOUND | JOURNAL DB2
CONTROL RECORD IS WRONG
VERSION**

Explanation:

The journal DB2 control record as not found or was not match the expected format. If the format is the problem, the record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22113E **UNKNOWN RECORD TYPE READ
FROM: tttttt; LOC=llll**

Explanation:

An unexpected record was found in the dataspace. The record is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22114I **JOURNAL LOOKUP KEY:
lookup_key**

Explanation:

This message accompanies CKZ11209E and indicates the key of the record, when an unexpected error occurs.

User response:

No action is required.

CKZ22115E **TARGET VOLUME IN DDSC
JOURNAL RECORD FOR DSN:
data_set_name IS ABSENT OR
INVALID**

Explanation:

Invalid target volume information was found for the data set name listed in the message, in record type DDSC, used with COPY-BY-DS. Processing stops and a return code of 8 is issued.

User response:

If unable to determine the cause of this error, contact IBM Software Support.

CKZ22116W **DDSC JOURNAL RECORD NOT
FOUND FOR DSN: data_set_name
-- CANNOT UPDATE ARCHIVE LOG
RECORD**

Explanation:

This message is a warning that is issued when the journal record type DDSC (used with COPY-BY-DS) for the given data set name is not found. This might occur when, for example, the dsname is an archive log that no longer exists.

User response:

Verify the data set name. If the data set exists, save the job listing and contact IBM Software Support.

CKZ22130I PROCESSING BSDSnn**Explanation:**

The indicated BSDS is being processed.

User response:

No action is required.

CKZ22135E BSDS CONTROL RECORD NOT FOUND**Explanation:**

The control record was not found in the BSDS. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22136W BSDS DB2 CATALOG DOES NOT MATCH KEYWORD**Explanation:**

The DB2 catalog alias found in the BSDS does not match a source name specified in the DB2-HLQS keyword. Processing continues.

User response:

Correct the DB2-HLQS name(s) specified if they are in error. If DB2UPDATE is being rerun and has already successfully updated the BSDS, the message may be ignored.

CKZ22137E BSDS RECORD NOT FOUND FOR recordtype**Explanation:**

An expected BSDS record type was not found. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22138W BSDS ACTIVE LOG # DSN: dsname DOES NOT MATCH RENAME MASKS**Explanation:**

An active log data set name does not match the rename masks specified during Db2 Cloning Tool RENAME. Processing continues.

User response:

The active log data set(s) should be on the volumes copied by Db2 Cloning Tool COPY and the old/new names specified in Db2 Cloning Tool RENAME. If DB2UPDATE is being rerun and has already successfully updated the BSDS, the message may be ignored.

CKZ22139W BSDS ARCHIVE LOG # DSN: dsname VOL: volser errortext**Explanation**

One of the following errors occurred:

- The archive log data set name does not match the rename masks specified during RENAME
- The archive log was not on a volume copied by Db2 Cloning Tool COPY
- For COPY-BY-DS, the source volume does not match.

Processing continues.

User response:

Check the details of the message to determine whether the job should be rerun with different control cards.

CKZ22140I BSDS UPDATED RECORD FOR recordtype**Explanation:**

A simulation of DB2UPDATE was requested. The updated BSDS record is printed, but, not rewritten.

User response:

No action is required.

CKZ22141W BSDS DSN: datasetname DOES NOT MATCH RENAME MASKS**Explanation:**

A BSDS data set name in the BSDS was not renamed due to there being no RENAME-MASKS entry that matches it.

User response:

Correct the RENAME-MASKS specification for Db2 Cloning Tool RENAME if they are in error.

CKZ22142E ERROR IN CHECKPOINT QUEUE RECORD OF BSDS0x description**Explanation:**

The checkpoint queue record was found but the record contains an error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22143E UNABLE TO RENAME DSN=datasetname USING MASK=target rename mask**Explanation:**

The new name of a data set will exceed 44 characters. Processing terminates.

User response:

Correct the RENAME-MASKS specification for Db2 Cloning Tool RENAME.

**CKZ22144E ERROR DETERMINING TARGET
VOLUME SERIAL LENGTH;
SOURCE: volser**

Explanation:

The program was unable to determine the length of the target volume serial. The volume pair entry is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22150W THE BSDS HAS A DDF RECORD
BUT THE DDF KEYWORD WAS NOT
SPECIFIED**

Explanation:

The BSDS contains a DDF record but the DDF keyword was not specified. Processing continues.

User response:

Add a DDF keyword if the DDF values should be changed.

**CKZ22151W BSDS DB2 CATALOG MATCHES
TARGET VALUE**

Explanation:

The DB2 catalog alias found in the BSDS matches a target name specified in the DB2-HLQS keyword. Processing continues.

User response:

Correct the DB2-HLQS name(s) specified if they are in error. If DB2UPDATE is being rerun and has already successfully updated the BSDS, the message may be ignored.

**CKZ22152W MEMBER: membername DOES
NOT MATCH DB2-MEMBERS
MASKS**

Explanation:

A member name in the BSDS was not renamed due to there being no DB2-MEMBERS entry that matches it. Processing continues.

User response:

Correct the DB2-MEMBERS name(s) specified if they are in error.

**CKZ22153W THE BSDS IS DATA SHARING BUT
THE DB2-GROUP AND DB2-
MEMBERS KEYWORDS WERE NOT
SPECIFIED**

Explanation:

The BSDS for this DB2 system is enabled for data sharing but the data sharing related keywords DB2-GROUP and DB2-MEMBERS were not specified. Processing continues.

User response:

Add the DB2-GROUP and DB2-MEMBERS keywords.

**CKZ22154W THE BSDS IS NOT DATA SHARING
BUT THE DB2-GROUP AND DB2-
MEMBERS KEYWORDS WERE
SPECIFIED**

Explanation:

The BSDS for this DB2 system is not enabled for data sharing but the data sharing related keywords DB2-GROUP and DB2-MEMBERS were specified. Processing continues.

User response:

Remove the DB2-GROUP and DB2-MEMBERS keywords.

**CKZ22155W THE DDF ALIAS KEYWORD WAS
SPECIFIED BUT THE DDF RECORD
IS NOT V8 FORMAT**

Explanation:

The ALIAS keyword was included in the DDF keyword but the DDF record in the BSDS indicates it is for a release prior to DB2 Version 8 that does not support ALIAS. Processing continues.

User response:

Remove the ALIAS keyword.

**CKZ22156W THE DDF keyword KEYWORD WAS
SPECIFIED BUT THE DDF RECORD
IS NOT V9 FORMAT**

Explanation:

The keyword was included in the DDF keyword but the DDF record in the BSDS indicates it is for a release prior to DB2 Version 9.1 that does not support the keyword. Processing continues.

User response:

Remove the keyword.

**CKZ22157W THE DDF ALIAS SECPORT VALUE
WAS SPECIFIED BUT THE DDF
RECORD IS NOT V9 FORMAT**

Explanation:

The ALIAS secpport value was included in the DDF keyword but the DDF record in the BSDS indicates it is for a release prior to DB2 Version 9.1 that does not support it. Processing continues.

User response:

Remove the secpport value.

**CKZ22158I A BACKUP SYSTEM RECORD IS IN
THE BSDS BUT IT HAS NO
ENTRIES**

Explanation:

A Backup System record was found in the BSDS and it contained no entries.

User response:

No action is required.

CKZ22160E SVC 99 INFO RETRIEVAL FAILED; DDNAME=ddname ERROR=xxxx INFO=xxxx

Explanation:

A problem occurred using SVC 99 information retrieval to get data set information about the data set allocated to DD ddname. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ22201I hh:mm:ss LINEAR FILE UPDATING STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss LINEAR FILE UPDATING COMPLETED; RETURN CODE=nnn RECORD COUNT=nnn**

Explanation:

DB2UPDATE processing message.

User response:

No action is required.

CKZ22204E OPEN FAILED FOR DDNAME=ddname

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ22207W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ22210E ERROR ACCESSING LINEAR FILE; DDNAME=ddname R15=nnnn ERROR=nnn FUNCTION=function LOC=lllll RBA OF RECORD: nnnnnnnnnnnn X'nnnnnnnn'

Explanation:

A VSAM error occurred accessing the indicated file. Processing terminates.

User response:

If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22212E UNABLE TO ESTABLISH ESTAEX; R15=nnnn

Explanation:

The program was not able to establish an estaex environment. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22230I PROCESSING DDNAME: ddname

Explanation:

The indicated linear file is being processed.

User response:

No action is required.

CKZ22231I THE FILE'S ENDING RBA IS: nnnnnnnnnnnn X'nnnnnnnn'

Explanation:

The ending rba of the linear file being processed.

User response:

No action is required.

CKZ22232I NO DBD INFORMATION FOUND

Explanation:

No DBD information was found. This can happen when the DBD01 table space is from a DB2 10 system and the DBD information has been moved to the SYSDBDXA table space.

User response:

No action is required.

CKZ22240I UPDATED, RECORD NUMBER: rrr FIELDS CHANGED: fff |CHILD RECORD UPDATED

Explanation:

A simulation of DB2UPDATE was requested. A record in the indicated file matched on DB2-HLQS, STOGROUPS, and/or volume serial number. The record is not rewritten. rrr is the relative record number in the file. fff is the number of occurrences of a field that was changed. 'CHILD RECORD UPDATED' indicates that the record had other entries associated with it.

User response:

No action is required.

CKZ22241E RECORD FLAGGED FOR CHANGE, BUT, NO CHANGE WAS SUCCESSFUL

Explanation:

The initial check indicated a record needed to be updated, but, the subsequent check did not find anything to update. Should not occur. The record is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22242I ddname **CHANGED RECORDS:**
nnnn **CHANGED FIELDS: ffff**

Explanation:

The linear file had nnnn records updated; ffff fields in those records were modified.

User response:

No action is required.

CKZ22243W ddname **INCONSISTENT RECORD**
SKIPPED: RECORD NUMBER: nnnn

Explanation:

An entry in the indicated file had was flagged as inconsistent. No changes will be made to this record.

User response:

No action is required.

CKZ22244E ddname **UNKNOWN SEGMENT;**
RECORD NUMBER: nnnn

Explanation:

A record part in the indicated file was not recognized. The first 16 bytes of recognized segments are printed; all of the unknown segment is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22245W ddname **DBD IS NOT TYPE 2039;**
RECORD NUMBER: nnnn

Explanation:

A record part in the indicated file was set as a DBD. The type should be 2039 but is not. The unknown DBD is printed. Processing continues.

User response:

No action is required.

CKZ22246E ddname **ERROR PARSING RECORD**
NUMBER: nnnn

Explanation:

The ESTAEX routine was entered while processing the indicated record. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22247E **INVALID RBA CALCULATED:**
nnnnnnnnnnnn X'nnnnnnnn'

Explanation:

The RBA calculated for a required record is greater than the end RBA for the file. The record that caused the RBA calculation is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22250W **VCAT NAME vcatname NOT**
CHANGED, IT DOES NOT MATCH
KEYWORD

Explanation:

A VCAT found in DBD01 does not match a source name specified in the DB2-HLQS keyword. Processing continues.

User response:

Correct the DB2-HLQS name(s) specified if they are in error.

CKZ22260E **TARGET STORAGE GROUP NAME**
MUST BE THE SAME LENGTH AS
THE SOURCE STORAGE GROUP
NAME | TARGET STORAGE GROUP
NAME: target-sgname | SOURCE
STORAGE GROUP NAME: source-
sgname | TARGET STORAGE
GROUP MASK: target-mask

Explanation:

The target storage group name is not the same length as the source storage group name. For DB2 Version 8 and later, the names must be the same length. Processing terminates.

User response:

Correct the values specified so they are the same length.

CKZ22261W **STORAGE GROUP NAME TOO LONG**
storage-group-name

Explanation:

The storage groupname found in DBD01 is longer than 30 characters. Processing continues.

User response:

This storage group name must be changed manually by use of the appropriate DB2 statements.

CKZ22262E **TARGET STORAGE GROUP NAME**
TOO LONG; MAX LENGTH: length |
TARGET STORAGE GROUP NAME:
target-sgname | SOURCE STORAGE
GROUP NAME: source-sgname |
TARGET STORAGE GROUP MASK:
target-mask

Explanation:

The target storage group name exceeds the maximum allowable length. Processing terminates.

User response:

Correct the values specified so the target is not greater than the indicated maximum length.

CKZ22299E ABEND DURING LINEAR FILE UPDATE

Explanation:

An abend occurred during the DBD01 update. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22301I hh:mm:ss DB2 XCF CLEAN UP STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss DB2 XCF CLEAN UP COMPLETED; RETURN CODE=nnn**

Explanation:

DB2UPDATE processing message.

User response:

No action is required.

CKZ22307W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ22310E UNEXPECTED RECORD TYPE ENCOUNTERED: type LOC=lllll

Explanation:

An unexpected condition was encountered while processing an XCF structure list.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22311E ERROR CALLING IXCQUERY; RC: rrr RSN: ssssssss LOC=lllll | ERROR CALLING IXLFORCE; RC: rrr RSN: ssssssss LOC=lllll

Explanation:

An error was received from the system macro call.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22312E ALLOCATED STRUCTURE TABLE IS FULL | CONNECTED STRUCTURE TABLE IS FULL | BAD STATUS STRUCTURE TABLE IS FULL

Explanation:

An internal table is too small.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22313E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ22314E JOURNAL DB2 MEMBER RECORD NOT FOUND

Explanation:

The journal DB2 member record was not found. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22315E JOURNAL DB2 MEMBER RECORD IS WRONG VERSION

Explanation:

The journal DB2 control record does not match the expected format. The record is printed. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22320I XCF STRUCTURES FOR DB2 DATA SHARING GROUP groupname: list of structures

Explanation:

This is a list of the DB2 XCF structures for the groupname.

User response:

No action is required.

CKZ22321E THE FOLLOWING XCF STRUCTURES HAVE AN UNDESIRABLE STATUS: list of structures

Explanation:

This is a list of the DB2 XCF structures that have a status that will not allow the clean up of the structures.

User response:

Verify that there are no target DB2 systems running.

CKZ22322I THE FOLLOWING XCF STRUCTURES WILL HAVE FAILED CONNECTIONS REMOVED: | THE FOLLOWING XCF STRUCTURES SHOULD HAVE FAILED CONNECTIONS REMOVED: list of structures

Explanation:

This is a list of the DB2 XCF structures that have failed connections that, (1) will be removed, or (2) should be removed.

User response:

(1) None. (2) The list of structures should have their failed connections removed manually.

CKZ22323I THE FOLLOWING XCF STRUCTURES WILL BE DEALLOCATED: | THE FOLLOWING XCF STRUCTURES SHOULD BE DEALLOCATED: list of structures

Explanation:

This is a list of the DB2 XCF structures that, (1) will be deallocated, or (2) should be deallocated.

User response:

(1) None. (2) The list of structures should be manually deallocated.

CKZ22331I XCF STRUCTURES NOT CLEANED UP DUE TO ERRORS

Explanation:

Due to errors all the DB2 XCF structures have not been cleaned up. The specific errors are described in other messages.

User response:

No action is required.

CKZ22332I ALL XCF STRUCTURES HAVE THE PROPER STATUS, NO CLEAN UP NECESSARY

Explanation:

The DB2 XCF structures are all in 'NOT ALLOCATED' status. There is no clean up necessary.

User response:

No action is required.

CKZ22333W NO XCF STRUCTURES FOUND FOR THIS DB2 SHARING GROUP

Explanation:

There were no XCF structures found for the DB2 sharing group.

User response:

No action is required.

CKZ22334I XCF STRUCTURES CLEANED UP

Explanation:

The XCF structures for the DB2 sharing group have been cleaned up.

User response:

No action is required.

CKZ22335I XCF STRUCTURES NOT CLEANED UP DUE TO SIMULATION

Explanation:

A simulation of DB2UPDATE was requested. The DB2 XCF structures were not cleaned up.

User response:

No action is required.

CKZ22336W XCF STRUCTURES SHOULD BE CLEANED UP BEFORE STARTING TARGET DB2

Explanation:

The DB2 XCF structures for the DB2 sharing group should be cleaned up prior to starting the target DB2.

User response:

Clean up the target DB2 XCF structures prior to starting the target DB2.

CKZ22337I XCF STRUCTURES CAN NOT BE CLEANED UP DUE TO ERRORS

Explanation:

A simulation of DB2UPDATE was requested. The DB2 XCF structures cannot be cleaned up due to errors. The specific errors are described in other messages.

User response:

Clean up the target DB2 XCF structures prior to starting the target DB2.

CKZ22340W XCF STRUCTURE DOES NOT HAVE EXPECTED DB2 STRUCTURE NAME, BYPASSED

Explanation:

The XCF structure name has a prefix of the target DB2 group name but does not follow the naming conventions for a DB2 XCF structure. The structure is bypassed from further processing.

User response:

No action is required.

CKZ22341W IXLFORCE REQUEST IS PENDING, THE SYSTEM WILL PROCESS IT WHEN THE DELAY CAUSE IS RESOLVED

Explanation:

The IXLFORCE request could not be processed immediately. The request is pending, and the system will process it when the condition causing the delay is resolved.

User response:

No action is required.

**CKZ22342E USER DOES NOT HAVE PROPER
SAF AUTHORIZATION TO CLEAN
UP THE XCF STRUCTURE**

Explanation:

The user running the job does not have SAF (RACF) authority to clean up the DB2 XCF structures.

User response:

Give the user running the job the proper authorization to allow clean up of the structures. Authorization is by entities with a format of IXLSTR.structurename in the FACILITY class.

**CKZ22350I XCF MEMBERS FOR DB2 DATA
SHARING GROUP groupname: list
of members**

Explanation:

A list of the members found for the groupname.

User response:

No action is required.

**CKZ22351I THE FOLLOWING XCF MEMBERS
HAVE AN UNDESIRABLE STATUS:
list of members**

Explanation:

The indicated members do not have quiesced or failed status.

User response:

The target DB2 system needs to be shut down if it is active.

**CKZ22351W THE FOLLOWING XCF MEMBERS
HAVE AN UNDESIRABLE STATUS:
list of members**

Explanation:

The indicated members do not have quiesced or failed status.

User response:

The target DB2 system needs to be shut down if it is active.

**CKZ22352I NO XCF MEMBERS FOUND FOR
DB2 DATA SHARING GROUP
groupname, NO CLEAN UP
NECESSARY**

Explanation:

No DB2 XCF members were found for the groupname.

User response:

No action is required.

CKZ22354I XCF MEMBERS CLEANED UP

Explanation:

The DB2 XCF members have been cleaned up.

User response:

No action is required.

**CKZ22355I XCF MEMBERS NOT CLEANED UP
DUE TO reason**

Explanation:

The DB2 XCF members have not been cleaned up. Reason indicates why the cleanup did not happen; SIMULATION if SIMULATE was specified, or DB2-XCFCLEAN(N) if DB2-XCFCLEAN(N) was specified.

User response:

No action is required.

**CKZ22357I XCF MEMBERS NOT CLEANED UP
DUE TO ERRORS**

Explanation:

The DB2 XCF members have not been cleaned up. Other messages will describe the problem encountered.

User response:

Check for other messages indicating the error which prevented the clean up.

**CKZ22358W THIS DB2 MEMBER
(membername) CAN NOT BE
STARTED DUE TO XCF MEMBERID
MISMATCH**

Explanation:

The start up of the target DB2 member will fail due to the memberid in the BSDS being different than the memberid in the XCF group member.

User response:

Do not start up the DB2 member.

**CKZ22359W XCF MEMBER membername NOT
CLEANED UP DUE TO ITS
UNDESIRABLE STATUS**

Explanation:

The XCF member membername was not cleaned up because it did not have quiesced or failed status. XCF members will only be cleaned up if they have quiesced or failed status.

User response:

The target DB2 system needs to be shut down if it is active. 25957

**CKZ22401I hh:mm:ss DIRECTORY UPDATING
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss
DIRECTORY UPDATING
COMPLETED; RETURN CODE=nnn**

Explanation:

DB2UPDATE processing message.

User response:

No action is required.

**CKZ22403I DDNAME=*dd-name* ALLOCATED
FOR DSN=*data-set-name***

Explanation:
ddname has been dynamically allocated for the specified data set.

User response:
No action is required.

**CKZ22404E OPEN FAILED FOR
DDNAME=*ddname***

Explanation:
'*ddname*' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:
If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22405E ALLOCATION FAILED FOR DSN:
*data-set-name***

Explanation:
Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:
If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ22405W DEALLOCATION FAILED FOR
DDNAME: *ddname***

Explanation:
Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

User response:
If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ22407W ERROR CALLING CKZ01HEX;
FUNCTION: *function* R15=nnnn**

Explanation:
An error occurred using CKZ01HEX to print a record. Processing continues.

User response:
Please report this message to IBM Software Support.

**CKZ22410E ERROR ACCESSING LINEAR FILE;
DDNAME=*ddname* R15=nnnn
ERROR=nnn FUNCTION=*function*
LOC=lllll RBA OF RECORD:**

**X'nnnnnnnn_nnnnnnnn' PAGE
NUMBER X'nnnnnnnn'**

Explanation:
A VSAM error occurred accessing the indicated file. Processing terminates.

User response:
If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22412E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn**

Explanation:
The program was not able to establish an *estaex* environment. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22423E SVC 99 INFORMATION RETRIEVAL
FAILED; DDNAME: *ddname* R15=
nnnn S99INFO= X'nnnn'
S99ERROR= X'nnnn'**

Explanation:
The SVC 99 information retrieval of the data set name of the identified DD has failed. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains these messages.

**CKZ22424E THE DATA SET NAME OF DDNAME:
ddname DOES NOT HAVE DB2
FORMAT; *data-set-name***

Explanation:
The data set name of the data set allocated to the indicated *ddname* does not have a DB2 data set name form. DB2UPDATE is trying to determine if there are multiple data sets for the DB2 directory LOB space SYSDBDXA that need to be processed, but cannot determine what the data set names should be. Processing terminates.

User response:
Change the JCL statement for the indicated *ddname* to point to the DB2 directory LOB space SYSDBDXA, which has a DB2 format data set name and also has a low level qualifier of A001.

**CKZ22425E THE NUMBER OF DIRECTORY
DATA SETS DISCOVERED IS
GREATER THAN THE EXPECTED
MAXIMUM OF 254**

Explanation:
More than 254 DB2 directory LOB space SYSDBDXA data sets have been discovered. A maximum of 254 data sets is expected. Processing terminates.

User response:

Verify the data set being used corresponds to the correct DB2 directory LOB space SYSDBDX data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22426I COMPRESSED DBD FOUND;
CATMAINT WILL BE USED TO
UPDATE DB2 DIRECTORY**

Explanation:

The DB2 directory LOB DSNDB01.SYSDBDXA has compressed entries. The CATMAINT utility will be used to update it.

User response:

No action is required.

CKZ22430I PROCESSING DDNAME: ddname

Explanation:

The indicated file is being processed.

User response:

No action is required.

**CKZ22431I THE ENDING RBA FOR DDNAME:
ddname IS:
X'nnnnnnnn_nnnnnnn'**

Explanation:

The ending RBA of the directory file being processed.

User response:

No action is required.

**CKZ22432E END OF FILE ENCOUNTERED
WHILE READING HEADER PAGE**

Explanation:

End of file was encountered while reading the header page. The data set allocated to the SYSDBDXA DD appears to have no data in it. Processing terminates.

User response:

Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22433E FIRST PAGE IS NOT LOB HEADER
PAGE**

Explanation:

The first page of the data set allocated to the SYSDBDXA DD is not a LOB header page. Processing terminates.

User response:

Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22434E INTERNAL ERROR; LOC=lllll
reason text**

Explanation:

An internal error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22435E UNEXPECTED PAGE TYPE
RETURNED FROM READ DIRECT**

Explanation:

Read direct got a page type that was not a LOB map or LOB header. The page is printed. Processing terminates.

User response:

Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22436W UNEXPECTED DBD IDENTIFIER;
MAP PAGE NUMBER: X'nnnnnnnn'**

Explanation:

The identifier of a DBD is not as expected. The DBD is skipped. The DBD is printed. Processing continues.

User response:

No action is required.

**CKZ22437W UNABLE TO UPDATE DBD DUE TO
POSSIBLY UNWRITTEN PAGES;
MAP PAGE NUMBER:
X'page_number' DATA PAGE
NUMBER: X'page_number'
DB2UPDATE WILL NEED TO RUN
AGAIN AFTER THE TARGET DB2 IS
STARTED AND RESOLVES THE
UNWRITTEN PAGES**

Explanation:

A page that is part of a DBD appears to be empty or has old data in it. This implies that possibly not all the pages of the DBD had been externalized by DB2 when the volumes were cloned. The updating of this DBD is bypassed. Processing continues.

User response:

DB2UPDATE with the DBD01ONLY keyword will need to be run after the target DB2 is started, DB2FIX DATABASES(DB2) has been run, and the target DB2 stopped. DB2FIX DATABASES(DB2) is also expected to issue message CKZ23526E when this happens.

**CKZ22440I UPDATED, RECORD STARTING ON
PAGE: X'nnnnnnnn' FIELDS
CHANGED: fff**

Explanation:

A simulation of DB2UPDATE was requested. A record in the file matched on DB2-HLQS, STOGROUPS, and/or volume serial number. The record is not rewritten. nnnnnnnn is the hex page number in the file. fff is the number of occurrences of a field that was changed.

User response:

No action is required.

CKZ22442I ddname **CHANGED RECORDS:**
nnnn **CHANGED FIELDS:** ffff

Explanation:

The directory file had nnnn records updated; ffff fields in those records were modified.

User response:

No action is required.

CKZ22443W ddname **INCONSISTENT PAGE;**
PAGE NUMBER: X'nnnnnnnn'
LOC=lllll

Explanation:

A page in the directory file was flagged as inconsistent. Processing continues.

User response:

No action is required.

CKZ22444E **PAGE READ IS NOT EXPECTED**
PAGE NUMBER; REQUESTED:
X'nnnnnnnn' READ: x'nnnnnnnn'
LOC=lllll

Explanation:

A direct read of a page did not return the requested page. The page is printed. Processing terminates.

User response:

Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22445E **CHECK BYTE ON PAGE IS NOT AS**
EXPECTED; EXPECTED: X'nn' GOT
x'nn' LOC=lllll

Explanation:

The check byte on a page does not have the expected value. The page is printed. Processing terminates.

User response:

Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22446E ddname **ERROR PARSING RECORD**
AT RBA: X'nnnnnnnn_nnnnnnnn'

Explanation:

The ESTAEX routine was entered while processing the indicated record. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ22450W **VCAT NAME vcatname NOT**
CHANGED, IT DOES NOT MATCH
KEYWORD

Explanation:

A VCAT found in the file does not match a source name specified in the DB2-HLQS keyword. Processing continues.

User response:

Correct the DB2-HLQS name(s) specified if they are in error.

CKZ22460E **TARGET STORAGE GROUP NAME**
MUST BE THE SAME LENGTH AS
THE SOURCE STORAGE GROUP
NAME | TARGET STORAGE GROUP
NAME: target-sgname | SOURCE
STORAGE GROUP NAME: source-
sgname | TARGET STORAGE
GROUP MASK: target-mask

Explanation:

The target storage group name is not the same length as the source storage group name. For DB2 Version 8 and later, the names must be the same length. Processing terminates.

User response:

Correct the values specified so they are the same length.

CKZ22461W **STORAGE GROUP NAME TOO LONG**
storage-group-name

Explanation:

The storage group name found in the directory is longer than 30 characters. Processing continues.

User response:

This storage group name must be changed manually by use of the appropriate DB2 statements.

CKZ22499E **ABEND DURING DIRECTORY**
UPDATE

Explanation:

An abend occurred during the directory update. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ223501I **hh:mm:ss DB2FIX STARTED -**
PROGRAM REV=rrr | hh:mm:ss
DB2FIX COMPLETED; RETURN
CODE=nnn

Explanation:

DB2FIX processing message.

User response:
No action is required.

**CKZ23506E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation:
A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ23507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:
An error occurred using CKZ01HEX to print a record. Processing continues.

User response:
Please report this message to IBM Software Support.

**CKZ23508E UNABLE TO LOAD PROGRAM:
program name**

Explanation:
The indicated program name was not found. Processing terminates.

User response:
Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ23520E DSNALI ERROR; FUNCTION=
function RC=nnnn RSN=nnnnnnnn**

Explanation:
An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response:
Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

**CKZ23521E IFI ERROR; FUNCTION= function
RC=nnnn RSN=nnnnnnnn**

Explanation:
An error occurred using the DB2 IFI interface. Function is the IFI function requested and RSN is the DB2 error reason. Processing terminates.

User response:
Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

**CKZ23522E DB2 SUBSYSTEM: db2ssid NOT
ACTIVE, RSN=00F30002 | DB2
SUBSYSTEM: db2ssid NOT
DEFINED, RSN=00F30006**

Explanation:
The DB2 subsystem is not active or not defined. Processing terminates.

User response:
Verify that the DB2 subsystem is active and the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ23523I CONNECT TO DB2 SUBSYSTEM:
ssid VERSION: nnn**

Explanation:
A connection has been established to the DB2 subsystem. Processing continues.

User response:
No action is required.

**CKZ23524I NO APPLICATION DATA OR INDEX
SPACES NEED TO BE STARTED |
NO DB2 DIRECTORY OR CATALOG
DATA OR INDEX SPACES NEED TO
BE STARTED**

Explanation:
No application data or index spaces have an LPL or GRECP status or no DB2 directory or catalog data or index spaces have an LPL or GRECP status. Processing continues.

User response:
No action is required.

**CKZ23525E THERE ARE DB2 DIRECTORY OR
CATALOG DATA OR INDEX SPACES
THAT NEED TO BE STARTED**

Explanation:
The request was for application databases but there are DB2 directory or catalog data or index spaces that have an LPL or GRECP status. Processing terminates.

User response:
Run DB2FIX with DATABASES(DB2) before running it with DATABASES(APPLICATION).

**CKZ23526E DSNDB01.space_name IS IN
RESTRICTED STATUS; DB2UPDATE
NEEDS TO RUN AGAIN**

Explanation:
The indicated space, DBD01 or SYSDBDXA, in database DSNDB01 had LPL or GRECP status. The starting of this space to fix the LPL or GRECP status has possibly caused the regression of updates made to it by DB2UPDATE. Processing continues.

User response:

The Db2 subsystem needs to be stopped and DB2UPDATE run again with the DBD01ONLY keyword to redo the updates to DBD01 or SYSDBXA.

CKZ23527E MEMBER: member HOLDS LOCKS; IT MUST BE STARTED TO RELEASE THE LOCKS

Explanation:

The identified data sharing member was active during the cloning and holds locks against data or index spaces. Processing terminates.

User response:

Start the identified data sharing member to release the locks. Then rerun DB2FIX.

CKZ23528E ERRORS PREVENT STARTING DATASPACES; MANUAL STARTS ARE NECESSARY

Explanation:

DB2FIX is unable to start dataspace due to an error or condition that is beyond the capability of DB2FIX. Processing terminates.

User response:

Manually identify and start the dataspace. Then contact IBM Software Support. Have available the listing that contains this message.

CKZ23529W MEMBER: member HOLDS LOCKS; DB2FIX MAY NEED TO BE RUN ON IT

Explanation:

The identified data sharing member was active during the cloning and holds locks against table or index spaces. If the identified member has not been started, it needs to be started. If the identified member has been started, it may be necessary to also run DB2FIX on the identified member to resolve LPL or GRECP status. Processing continues.

User response:

If DB2FIX fails to resolve LPL or GRECP status, run DB2FIX on the identified data sharing member using ACTION(CONTINUE) in the MEMBERS-NEED-STARTING keyword.

CKZ23530W NON NORMAL STATUS; DATABASE: database SPACENAM: spacename PART: partition number STATUS: status

Explanation:

The identified space has a status that is restricted but does not include LPL or GRECP status. Processing continues.

User response:

This space will require manual action to remove the restricted status.

CKZ23531E STATUS PREVENTS STARTING; DATABASE: database SPACENAM: spacename PART: partition number STATUS: status

Explanation:

The identified space has a status that is restricted and has LPL or GRECP status and other restricted status. Processing terminates.

User response:

This space will require manual action to remove the LPL or GRECP status.

CKZ23532W UNKNOWN RESPONSE LINE

Explanation:

The response to a DB2 display command included an unknown line. Processing continues.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ23533E FORMAT OF RESPONSE LINE NOT AS EXPECTED

Explanation:

The response to a DB2 display command included a line that did not have the expected format. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ23534E USERID IS NOT AUTHORIZED TO ISSUE THE DB2 COMMAND

Explanation:

The userid running DB2FIX does not have the necessary authorization to issue the DB2 command. Processing terminates.

User response:

The userid running DB2FIX needs to be defined as an install SYSADM in the DB2 zparms.

CKZ23535E COMMAND FAILED; RC= nnn

Explanation:

The DB2 command failed to process. Nnn is the return code from the DB2 command processor. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated DB2 messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ23536E PAGE SPACE(S) DID NOT FIX WITHIN WAIT TIME LIMIT; DATABASE(dbname) SPACENAM(spname) clone

Explanation:

The DB2 subsystem did not complete the start command of the indicated page space(s) to resolve the LPL or GRECP status within the time limit specified in the WAIT keyword. Processing terminates.

User response:

Determine why the DB2 subsystem did not complete the start command. If the wait time limit is too short, increase it.

**CKZ23537I WAITING FOR PAGE SPACE(S) TO
FIX; DATABASE(dbname)
SPACENAM(spname) clone**

Explanation:

Db2 Cloning Tool is waiting for the DB2 subsystem to complete the start command of the indicated page space(s) to resolve the LPL or GRECP status. Processing continues.

User response:

No action is required.

**CKZ23538I PAGE SPACE(S) FIXED;
DATABASE(dbname)
SPACENAM(spname) clone**

Explanation:

The DB2 subsystem has completed the start command of the indicated page space(s) and resolved the LPL or GRECP status. Processing continues.

User response:

No action is required.

**CKZ23539E EXCESSIVE LOOPING;
PROCESSING TERMINATED**

Explanation:

Excessive looping has been detected while trying to fix application page and index spaces. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ23540I DB2 COMMAND: db2 command

Explanation:

Displays the command that will be executed by DB2. Processing continues.

User response:

No action is required.

**CKZ23541I COMMAND NOT EXECUTED DUE
TO SIMULATION MODE**

Explanation:

The DB2 command displayed in message CKZ23540I was not executed because this is a simulate run. Processing continues.

User response:

No action is required.

**CKZ23542I dbname.spname WILL NOT BE
STARTED DUE TO EXCLUDE-
MASKS ENTRY: exclude-mask**

Explanation:

The page space dbname.spname was found to have LPL or GRECP status but will not be started because there is an entry in the EXCLUDE-MASKS keyword it matches. Processing continues.

User response:

The page space is not started.

**CKZ23547E CKZ00900 UNEXPECTED
RESULTS; error text**

Explanation:

An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ23548I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation:

Informational message indicating how DB2FIX will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

**CKZ23550E ERROR IN PARAMETERS FOR
keyword**

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ23551E REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ23553E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the

maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ23554E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ23556E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ23558E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ23585I VALUES FOR KEYWORD: keyword
list of values**

Explanation:

Parsing found the listed values for the keyword.

User response:

No action is required.

CKZ23586I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

CKZ23590E INTERNAL ERROR; LOC=lllll

Explanation:

An internal processing error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24001I hh:mm:ss DB2RBLDBSDS
STARTED - PROGRAM REV=rrr |**

**hh:mm:ss DB2RBLDBSDS
COMPLETED; RETURN CODE=nnn**

Explanation:

DB2RBLDBSD processing message.

User response:

No action is required.

**CKZ24003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ24004E DDNAME MISSING: ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

**CKZ24005E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ24008E UNABLE TO LOAD PROGRAM:
 program name**

Explanation:

The indicated program name was not found.
Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ24009E ERROR ACCESSING JOURNAL
 FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ24011E JOURNAL CONTROL RECORD NOT
 FOUND | JOURNAL DB2 CONTROL
 RECORD NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24012E JOURNAL DB2 CONTROL RECORD
 IS WRONG VERSION**

Explanation:

The journal DB2 control record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ24020E NO DB2 GROUP NAME

Explanation:

A DB2 group name was not found in the journal DB2 control record.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ24021E NO DB2 CATALOG NAME

Explanation:

A DB2 catalog name was not found in the journal DB2 control record.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24030I OPTIONS IN EFFECT FOR THIS
 EXECUTION: merged options**

Explanation:

Informational message indicating how DB2RBLDBSDS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

**CKZ24051E REQUIRED KEYWORD MISSING:
 keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ24053E KEYWORD: keyword MAXIMUM
 LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ24054E KEYWORD HAS MORE THAN 1
 OPERAND; ONLY 1 ALLOWED:
 keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ24056E NOTHING SPECIFIED FOR
 KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ24058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ24101I hh:mm:ss BSDS REBUILD
STARTED - PROGRAM REV=rrr |
hh:mm:ss BSDS REBUILD
COMPLETED; RETURN CODE=nnn**

Explanation:

BSDS REBUILD processing message.

User response:

No action is required.

**CKZ24103I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ24104E OPEN FAILED FOR
DDNAME=ddname RC=nnn VSAM
RC=nnn**

Explanation:

Open has failed for ddname. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24105E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ24109E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ24111E JOURNAL DB2 MEMBER LOG
RECORD NOT FOUND | JOURNAL
DB2 MEMBER RECORD NOT
FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24112E JOURNAL DB2 MEMBER LOG
RECORD IS WRONG VERSION |
JOURNAL DB2 MEMBER RECORD
IS WRONG VERSION | JOURNAL
DB2 DDF RECORD IS WRONG
VERSION**

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ24120E ERROR ACCESSING BSDS FILE;
DDNAME=ddname LOC=location**

Explanation:

A VSAM error occurred accessing a BSDS file. Processing terminates.

User response:

See the associated CKZVSEnnE error messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ24121E BSDS CONTROL RECORD NOT FOUND

Explanation:

The control record was not found in the BSDS. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ24122E BSDS CONTROL RECORD INCORRECT

Explanation:

The control record that was found in the BSDS appears to be incorrect. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ24123I DSN=*data_set_name1* COPIED TO DSN=*data_set_name2*

Explanation:

The contents of the *data_set_name1* have been copied to *data_set_name2*.

User response:

No action is required.

CKZ24130E DB2UPDATE HAS NOT BEEN RUN FOR THIS MEMBER

Explanation:

DB2UPDATE has not been run for this member. processing terminates.

User response:

Run DB2UPDATE for this member.

CKZ24131E DB2UPDATE FOR THIS MEMBER WAS SIMULATION, SIMULATION MUST BE SPECIFIED

Explanation:

Simulation was not specified but the DB2UPDATE run for this member was a simulation.

User response:

Run with simulation specified.

CKZ24135E DDNAME=*ddname* IS ALREADY ALLOCATED

Explanation:

The ddname is already allocated and it must not be.

User response:

Ensure the ddname is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ24136I DSN=*datasetname* HAS BEEN CLEARED

Explanation:

All data has been removed from the data set.

User response:

No action is required.

CKZ24140E UTILITY EXECUTION FAILED

Explanation:

The invoked utility has failed.

User response:

: If unable to determine the reason for the failure from the associated messages, contact IBM Software Support. Have available the listing that contains this message.

CKZ24501I hh:mm:ss DB2SQL STARTED - PROGRAM REV=rrr | hh:mm:ss DB2SQL COMPLETED; RETURN CODE=nnn

Explanation:

DB2SQL processing message.

User response:

No action is required.

CKZ24503I DDNAME=*ddname* ALLOCATED FOR DSN=*datasetname*

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ24504E DDNAME MISSING: *ddname*

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ24505E ALLOCATION FAILED FOR DSN: *datasetname*

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ24508E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ24509E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ24511E JOURNAL record_type NOT FOUND

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. The *record_type* listed in the message is the type of record that is not found. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE

command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24512E JOURNAL record_type IS WRONG
VERSION**

Explanation:

The journal record does not match the expected format. The incorrect *record_type* is listed in the message. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ24513E COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc LOC=lllll**

Explanation:

There was a problem with the journal records needed to initiate the command. The number of records read from the journal, rrrr, is not the same as the number indicated in the journal DB2 control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ24514I COPY-BY-DS JOURNAL FOUND

Explanation:

The journal data set was created by COPY-BY-DS. Processing continues.

User response:

No action is required.

**CKZ24520E DSNALI ERROR; FUNCTION=
function RC=nnnn RSN=nnnnnnnn**

Explanation:

An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

**CKZ24522E DB2 SUBSYSTEM: db2ssid NOT
ACTIVE, RSN=00F30002 | DB2
SUBSYSTEM: db2ssid NOT
DEFINED, RSN=00F30006**

Explanation:

The DB2 subsystem is not active or not defined. Processing terminates.

User response:

Verify that the DB2 subsystem is active and the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ24523I CONNECT TO DB2 SUBSYSTEM:
 ssid VERSION: nnn**

Explanation:

A connection has been established to the DB2 subsystem. Processing continues.

User response:

No action is required.

**CKZ24524E PLAN: planname NOT USABLE OR
 MAY NOT EXIST, RSN=00F30040**

Explanation:

A bind for this plan has not been done or the plan is not usable. Processing terminates.

User response:

Bind the Db2 Cloning Tool plan to the DB2 subsystem.

**CKZ24548I OPTIONS IN EFFECT FOR THIS
 EXECUTION: merged options**

Explanation:

Informational message indicating how DB2SQL will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

**CKZ24540E CKZ00900 UNEXPECTED
 RESULTS; error text**

Explanation:

An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24550E ERROR IN PARAMETERS FOR
 keyword**

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ24551E REQUIRED KEYWORD MISSING:
 keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ24552E REQUIRED INI SECTION/TOKEN
 MISSING: SECTION=section
 TOKEN=token | REQUIRED INI
 VALUE MISSING FOR
 SECTION=section TOKEN=token |
 INVALID INI VALUE FOR
 SECTION=section TOKEN=token**

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

**CKZ24553E KEYWORD: keyword MAXIMUM
 LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ24554E KEYWORD HAS MORE THAN 1
 OPERAND; ONLY 1 ALLOWED:
 keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ24556E NOTHING SPECIFIED FOR
 KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ24558E INVALID VALUE IN KEYWORD:
 keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ24575E UNMATCHED ENTRIES IN
 KEYWORD: keyword**

Explanation:

For WLM-ENVIRONMENT-MASKS there must be pairs of source and target masks. An uneven number of entries was specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ24576E THE USE OF keyword REQUIRES DB2 V9

Explanation:

The keyword can only be used with DB2 Version 9.1 and the DB2 system is less than DB2 Version 9.1. Processing terminates.

User response:

Remove the keyword.

CKZ24577W The entries in keyword will not be used because the DB2 directory LOB is not compressed

Explanation:

The indicated keyword was specified in the DB2SQL command, but the DB2 directory LOB is not compressed. The keyword is only used when the DB2 directory LOB is compressed.

User response:

Remove the keyword from the DB2SQL command.

CKZ24585I PAIRS FOR KEYWORD: keyword list of pairs

Explanation:

Parsing found the listed pairs for the keyword.

User response:

No action is required.

CKZ24586I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

CKZ24601I hh:mm:ss SQL PROCESSOR STARTED - PROGRAM REV=rrr | hh:mm:ss SQL PROCESSOR COMPLETED; RETURN CODE=nnn

Explanation:

SQL PROCESSOR processing message.

User response:

No action is required.

CKZ24606E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the PARMLIB member that controls execution of Db2 Cloning Tool.

CKZ24607W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ24620W DSNTIAR FAILED; RC:nnnn RSN: nnnnnnnn

Explanation:

An error occurred using DSNTIAR to format error messages. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ24621E SQL function FAILED, SQLCODE: sql_code SQLSTATE: sql_state

Explanation:

An error occurred processing an SQL statement. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24623W NON SMS-MANAGED STOGROUP CANNOT BE CLONED TO TARGET: storage_group WITH VOLID (volume)

Explanation:

Specific volume IDs in SYSVOLUMES are not allowed with COPY-BY-DS. The storage group is not cloned. Processing continues.

User response:

Check the details of the message to determine whether the job should be rerun with different control cards.

CKZ24624I SKIPPING SYSVOLUMES UPDATES (COPY-BY-DS)

Explanation:

The SYSVOLUMES VOLID does not need to be updated with COPY-BY-DS, because only VOLID='*' can be cloned. Processing continues.

User response:

No action is required.

CKZ24630I **type INFORMATION UPDATED IN TABLE: tablename CHANGED ROWS: nnnnnn | type INFORMATION WOULD HAVE BEEN UPDATED IN TABLE: tablename CHANGED ROWS: nnnnnn**

Explanation:

The table has been updated, or would have been updated, with the indicated information type. Processing continues.

User response:

No action is required.

CKZ24631I **text**

Explanation:

Displays the generated SQL statement being processed. Processing continues.

User response:

No action is required.

CKZ24632I ***New / Old DB2 STOGROUPS created / dropped: number_created_or_dropped | New / Old DB2 STOGROUPS would have been created / dropped: number_created_or_dropped***

Explanation:

The new name DB2 STOGROUPS have been or would have been created, or the old name DB2 STOGROUPS have been or would have been dropped. Processing continues.

User response:

No action is required.

CKZ24633I ***New / Old DB2 STOGROUP already created | dropped: stogroup-name .***

Explanation:

New name DB2 STOGROUPS have already been created, or old name DB2 STOGROUPS have already been dropped. Processing continues.

User response:

No action is required.

CKZ24634I ***Volume volser in STOGROUP stogroup-name is not a source volume New STOGROUP new-stogroup-name***

Explanation:

A volser in DB2 STOGROUP *stogroup-name* is not a source volume of the cloning. The new DB2 STOGROUP *new-stogroup-name* being created will have this volume. Processing continues.

User response:

No action is required. You might want to verify that the volser is intended to be used in the DB2 STOGROUP. Manually remove the volume from the DB2 STOGROUP if desired.

CKZ24640I **DB2 catalog level: level**

Explanation:

This message shows the current DB2 catalog level.

User response:

No action is required.

CKZ24641W **DB2 catalog level has unexpected form: xxx**

Explanation:

The DB2 catalog level retrieved from DB2 does not have the expected form.

User response:

Report this message to IBM Software Support.

CKZ24645W **VCAT NAME *vcatname* NOT CHANGED, IT DOES NOT MATCH KEYWORD**

Explanation:

A VCAT found in the DB2 catalog does not match a source name that is specified in the DB2-HLQS keyword in the prior DB2UPDATE command or in the DB2-HLQS keyword specified in the DB2SQL command.

User response:

Correct the DB2-HLQS names in the DB2SQL command if they are in error.

CKZ24650W **item value NOT CHANGED, IT DOES NOT MATCH KEYWORD**

Explanation:

For item of WLM ENVIRONMENT, the WLM_ENVIRONMENT value found in SYSIBM.SYSROUTINES did not match a source mask specified in the WLM-ENVIRONMENT-MASKS keyword. For item of DATACLAS, the DATACLAS value found in SYSIBM.SYSSTOGROUP did not match a source mask specified in the DATACLAS-MASKS keyword. For item of MGMTCLAS, the MGMTCLAS value found in SYSIBM.SYSSTOGROUP did not match a source mask specified in the MGMTCLAS-MASKS keyword. For item of STORCLAS, the STORCLAS value found in SYSIBM.SYSSTOGROUP did not match a source mask specified in the STORCLAS-MASKS keyword. Processing continues.

User response:

Correct the WLM-ENVIRONMENT-MASKS, DATACLAS-MASKS, MGMTCLAS-MASKS, or STORCLAS-MASKS names if they are in error.

CKZ24651W WLM ENVIRONMENT NOT CHANGED FOR procedure-name UNABLE TO ESTABLISH NEEDED ENVIRONMENT SETTINGS FOR ALTER**Explanation:**

An attempt to change the WLM ENVIRONMENT name for a native SQL procedure failed. The ALTER procedure SQL statement being used requires that the environment settings in effect must be the same as when the procedure was first created. Db2 Cloning Tool changes the settings of CURRENT PATH and CURRENT SCHEMA to match what they were at the time the procedure was created. Other environment settings can come from system definitions such as the DSNHDECP module. If these other environment settings do not match what they were at the time the procedure was created the ALTER will fail with a SQLCODE of -4706. A DSNT408I message should be displayed that shows the ENVID of the environment that existed at the time of the ALTER and the ENVID of the environment that existed when the procedure was created. Processing continues.

User response:

The WLM ENVIRONMENT name for this procedure will need to be changed manually. The ENVIDs in the DSNT408I message can be used for selecting the corresponding rows from SYSIBM.SYSENVIRONMENT. Comparing the two rows will show which environment settings are different and need to be set for an ALTER statement to work for this procedure.

CKZ24652W ENVID=nnn NOT FOUND IN SYSIBM.SYSENVIRONMENT; UNABLE TO SET ENVIRONMENT FOR ALTER**Explanation:**

The row in SYSIBM.SYSENVIRONMENT for the ENVID of a native SQL procedure was not found. This condition could indicate a problem with the DB2 catalog and tables SYSIBM.SYSENVIRONMENT and SYSIBM.SYSROUTINES. There should be a following CKZ24651W message that will identify the procedure name. Processing continues.

User response:

The WLM ENVIRONMENT name for this procedure will need to be changed manually. The reason for the missing row in SYSIBM.SYSROUTINES should be determined and corrected.

CKZ24653I ALTER RECEIVED SQLCODE: -449; WILL ADD EXTERNAL NAME AND RETRY ALTER**Explanation:**

The ALTER statement received a SQLCODE -449. This SQLCODE indicates that EXTERNAL NAME must be

included in the ALTER statement. EXTERNAL NAME will be added to the ALTER statement and the modified ALTER will be executed.

User response:

No action is required.

CKZ24701I hh:mm:ss INVOKE DSNUTILB STARTED - PROGRAM REV=revision | hh:mm:ss INVOKE DSNUTILB; RETURN CODE=return_code**Explanation:**

This message is issued when DSNUTILB is invoked.

User response:

No action is required.

CKZ24703I DDNAME=ddname ALLOCATED FOR DSN=data_set_name**Explanation:**

ddname has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ24704E OPEN FAILED FOR DDNAME=ddname**Explanation:**

Open has failed for ddname. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ24705E ALLOCATION FAILED FOR DSN: data_set_name**Explanation:**

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24705W DEALLOCATION FAILED FOR DDNAME: ddname**Explanation:**

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ24707W ERROR CALLING CKZ01HEX;
FUNCTION=function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

**CKZ24733E Unable to change DB2 STOGROUP
name for DATABASE= database**

Explanation:

A STOGROUP name was found for an object in database DSNDB01, DSNDB06, or DSNDB07. This situation should not occur. REPAIR DBD does not allow these data bases to be specified.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ24734E INTERNAL ERROR; reason

Explanation:

An internal error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24735E DDNAME=ddname IS ALREADY
ALLOCATED**

Explanation:

The *ddname* is already allocated; this *ddname* must not already be allocated.

User response:

Ensure that the *ddname* is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

**CKZ24740E DSNUTILB UTILITY EXECUTION
FAILED**

Explanation:

The invoked utility failed.

User response:

If unable to determine the reason for the failure from the associated messages, contact IBM Software Support. Have available the listing that contains this message.

**CKZ24760E ATTACH ERROR; RETURN CODE =
return_code REASON CODE =
reason_code**

Explanation:

An attempt to attach a subtask failed with the return code and reason code that are listed in the message.

User response:

Ensure that the DB2 runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ24801I hh:mm:ss ISSUE DB2 COMMANDS
STARTED -PROGRAM REV=revision
| hh:mm:ss ISSUE DB2
COMMANDS; RETURN
CODE=return_code**

Explanation:

This message is issued when processing DB2 commands.

User response:

No action is required.

**CKZ24807W ERROR CALLING CKZ01HEX;
FUNCTION=function R15=nnnn**

Explanation:

An error occurred when using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

CKZ24820E INTERNAL ERROR; reason

Explanation:

An internal error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24821E IFI ERROR; FUNCTION= function
RC=return_code RSN=reason_code
LOC=location**

Explanation:

An error occurred using the DB2 IFI interface. *function* is the IFI function that was requested; *reason_code* is the DB2 error reason; *location* is the location where the error occurred. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

**CKZ24834E USERID IS NOT AUTHORIZED TO
ISSUE THE DB2 COMMAND**

Explanation:

The user ID that is running DB2SQL does not have the necessary authorization to issue the DB2 command. Processing terminates.

User response:

The user ID that is running DB2SQL must be defined as an install SYSADM in the DB2 zparms.

CKZ24840E DB2 COMMAND: db2_command

Explanation:

This message displays the command that will be executed by DB2. Processing continues.

User response:

No action is required.

CKZ24841I COMMAND NOT EXECUTED DUE TO SIMULATION MODE

Explanation:

The DB2 command that is displayed in message CKZ24840I was not executed because this is a simulate run.

User response:

No action is required.

CKZ25001I hh:mm:ss DB2LGRNXCLEAN STARTED - PROGRAM REV=rrr | hh:mm:ss DB2LGRNXCLEAN COMPLETED; RETURN CODE=nnn

Explanation:

DB2LGRNXCLEAN processing message.

User response:

No action is required.

CKZ25003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ25004E DDNAME MISSING: ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ25005E ALLOCATION FAILED FOR DSN: datasetname

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ25005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ25007W ERROR CALLING CKZ01HEX; FUNCTION=function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

CKZ25008E UNABLE TO LOAD PROGRAM: program name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ25009E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25011E JOURNAL type RECORD NOT FOUND

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25012E JOURNAL type RECORD IS WRONG VERSION

Explanation:

A journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. A journal created by a prior release of Db2 Cloning Tool can be upgraded to the current release by using the JRNLUPEGRADE command. See the “[JRNLUPEGRADE](#)” on page 470 topic for the scenarios where JRNLUPEGRADE can be used. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25020E NO DB2 CATALOG NAME

Explanation:

A DB2 catalog name was not found in the journal DB2 control record.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ25030I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how DB2RBLDBSDS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ25031E DB2UPDATES WERE ALL SIMULATIONS, SIMULATION MUST BE SPECIFIED

Explanation:

All DB2UPDATE runs were simulations so this must be a simulation. Processing terminates.

User response:

Run with simulation specified.

CKZ25051E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ25053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ25054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ25056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ25058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ25101I hh:mm:ss SYSLGRNX CLEAN STARTED - PROGRAM REV=rrr | hh:mm:ss SYSLGRNX CLEAN COMPLETED; RETURN CODE=nnn

Explanation:

SYSLGRNX CLEAN processing message.

User response:

No action is required.

CKZ25103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ25104E OPEN FAILED FOR DDNAME=ddname

Explanation:

Open has failed for ddname. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ25105E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ25105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ25107W ERROR CALLING CKZ01HEX;
FUNCTION=function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

**CKZ25108E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ25121E DDNAME=ddname IS ALREADY
ALLOCATED**

Explanation:

The ddname is already allocated and it must not be.

User response:

Ensure the ddname is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

**CKZ25122I SPACE=spacename
DSN=datasetname HAS BEEN
CLEANED**

Explanation:

All data has been removed from the data set.

User response:

No action is required.

CKZ25124I SPACE=spacename HAS attribute

Explanation:

The identified table or index space has the indicated attribute.

User response:

No response is required.

**CKZ25132I DATA FROM DDNAME=ddname IS
BEING USED**

Explanation:

Data is being used from the file allocated to ddname.

User response:

No action is required.

**CKZ25131E DSNUGICR UTILITY EXECUTION
FAILED**

Explanation:

The invoked DSNUGICR utility has failed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ25141E UNABLE TO DETERMINE IF
EXTENDED RBA IS ENABLED FOR
DSNDB01.tname.**

Explanation:

: It cannot be determined whether the indicated table or index space that is in the DB2 directory was enabled for extended RBA. It is not possible to properly clean the table or index space. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ25201I hh:mm:ss DB2UTILXCLEAN
STARTED - PROGRAM REV=rrr |
hh:mm:ss DB2UTILXCLEAN
COMPLETED; RETURN CODE=nnn**

Explanation:

This is an informational DB2UTILXCLEAN processing message.

User response:

No response is required.

**CKZ25203I DDNAME=ddname ALLOCATED
FOR DSN=data_set_name**

Explanation:

The ddname that is listed in the message text was dynamically allocated for the indicated data set.

User response:
No response is required.

CKZ25204E DDNAME MISSING: *ddname*

Explanation:
The ddname that is listed in the message text was specified for Db2 Cloning Tool to use, but is missing. Processing terminates.

User response:
Either correct the specified ddname, or add the appropriate ddname to the job's JCL.

**CKZ25205E ALLOCATION FAILED FOR DSN:
*data_set_name***

Explanation:
Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:
If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ25205W DEALLOCATION FAILED FOR
DDNAME: *ddname***

Explanation:
Dynamic deallocation for the ddname that is listed in the message failed. The associated z/OS messages are displayed. Processing continues.

User response:
If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ25207W ERROR CALLING CKZ01HEX;
FUNCTION: *function R15=nnnn***

Explanation:
An error occurred while CKZ01HEX was attempting to print a record. Processing continues.

User response:
If unable to determine the reason for the failure, contact IBM Software Support.

**CKZ25208E UNABLE TO LOAD PROGRAM:
*program_name***

Explanation:
The program name that is listed in the message text was not found. Processing terminates.

User response:
Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ25209E ERROR ACCESSING JOURNAL
FILE; LOC=*location***

Explanation:
A VSAM error occurred accessing a file. Processing terminates.

User response:
Refer to the associated CKZVSEnnE error messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ25211E JOURNAL *type* RECORD NOT
FOUND**

Explanation:
An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:
Verify that the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ25212E JOURNAL *type* RECORD IS WRONG
VERSION**

Explanation:
The identified journal record does not match the expected format. The record is printed. Processing terminates.

User response:
Verify that different releases of Db2 Cloning Tool were not run using the same journal data set. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25220E NO DB2 CATALOG NAME

Explanation:
A DB2 catalog name was not found in the journal DB2 control record.

User response:
Contact IBM Software Support. Have available the listing that contains these messages.

**CKZ25230I OPTIONS IN EFFECT FOR THIS
EXECUTION: *merged options***

Explanation:
This informational message indicates how DB2UTILXCLEAN will handle the options. The displayed options are derived from the INI file and any overriding specifications in the command input.

User response:
No action is required.

CKZ25231E DB2UPDATES WERE ALL SIMULATIONS, SIMULATION MUST BE SPECIFIED

Explanation:

All DB2UPDATE runs were simulations, so this run must be a simulation. Processing terminates.

User response:

Run with simulation (SIM) specified.

CKZ25251E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword that is required for processing was omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ25253E KEYWORD: keyword MAXIMUM LENGTH: length EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length that is allowed for the operand. *length* is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ25254E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is allowed. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ25256E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ25258E INVALID VALUE IN KEYWORD: keyword VALUE: value error_text

Explanation:

The value in the keyword that is listed in the message is invalid. *error_text* indicates the problem that was detected with the value. Processing terminates.

User response:

Correct the value that is specified in the keyword.

CKZ25301I hh:mm:ss SYSUTILX CLEAN STARTED - PROGRAM REV=rrr | hh:mm:ss SYSUTILX CLEAN COMPLETED; RETURN CODE=return_code

Explanation:

This message is a SYSUTILX CLEAN processing informational message.

User response:

No action is required.

CKZ25303I DDNAME=ddname ALLOCATED FOR DSN=data_set_name

Explanation:

ddname was dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ25304E OPEN FAILED FOR DDNAME=ddname

Explanation:

Open failed for the *ddname* that is listed in the message. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ25305E ALLOCATION FAILED FOR DSN: data_set_name

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25305W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25307W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred while CKZ01HEX was attempting to print a record. Processing continues.

User response:

If unable to determine the reason for the failure, contact IBM Software Support.

**CKZ25308E UNABLE TO LOAD PROGRAM:
 program_name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ25321E DDNAME=ddname IS ALREADY
 ALLOCATED**

Explanation:

The ddname is already allocated and it must not be.

User response:

Ensure that the ddname is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

**CKZ25322I SPACE=spacename
 DSN=data_set_name HAS BEEN
 CLEANED**

Explanation:

All data was removed from the data set.

User response:

No action is required.

**CKZ25323I DATA FROM DDNAME=ddname IS
 BEING USED**

Explanation:

Data is being used from the file allocated to *ddname*.

User response:

No action is required.

CKZ25324I SPACE=spacename HAS attribute

Explanation:

The identified table or index space has the indicated attribute.

User response:

No action is required.

**CKZ25331E DSNUGICR UTILITY EXECUTION
 FAILED**

Explanation:

The invoked DSNUGICR utility failed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ25341E UNABLE TO DETERMINE IF
 EXTENDED RBA IS ENABLED FOR
 DSNDB01.tsname**

Explanation:

It cannot be determined whether the indicated table or index space in the DB2 directory has been enabled for extended RBA. It is not possible to properly clean the table or index space. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ25401I hh:mm:ss EXTENDED RBA
 CHECKER STARTED - PROGRAM
 REV=rrr | hh:mm:ss EXTENDED
 RBA CHECKER COMPLETED;
 RETURN CODE=return_code**

Explanation:

Informational EXTENDED RBA CHECKER processing message.

User response:

No action is required.

**CKZ25403I DDNAME=ddname ALLOCATED
 FOR DSN=data_set_name**

Explanation:

The ddname that is listed in the message was dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ25405E ALLOCATION FAILED FOR DSN:
 data_set_name**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ25405W DEALLOCATION FAILED FOR
 DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software

Support. Have available the listing that contains these messages.

**CKZ25407W ERROR CALLING CKZ01HEX;
FUNCTION: *function* R15=*nnnn***

Explanation:

An error occurred while CKZ01HEX was attempting to print a record. Processing continues.

User response:

If unable to determine the reason for the failure, contact IBM Software Support.

**CKZ25410E ERROR ACCESSING LINEAR FILE;
DDNAME=*ddname* R15=*nnnn*
ERROR=*nnn* FUNCTION=*function*
LOC=*location* RBA OF RECORD:
X'*nnnnnnnnn_nnnnnnnn*'**

Explanation:

A VSAM error occurred accessing the indicated file. Processing terminates.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ25442I DSNDB01.*tsname*
DDNAME=*ddname* IS EMPTY**

Explanation:

The indicated table or index space in the DB2 directory is empty.

User response:

No action is required.

**CKZ25501I hh:mm:ss DB2START STARTED -
PROGRAM REV=*rrr* | hh:mm:ss
DB2START COMPLETED; RETURN
CODE=*nnn***

Explanation:

DB2START processing message.

User response:

No action is required.

**CKZ25507W ERROR CALLING CKZ01HEX;
FUNCTION: *function* R15=*nnnn***

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ25508E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ25520E DSNALI ERROR; FUNCTION=
function RC=*nnnn* RSN=*nnnnnnnnn***

Explanation:

An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

**CKZ25522E DB2 SUBSYSTEM: *db2ssid* NOT
DEFINED | DB2 SUBSYSTEM:
db2ssid NOT DEFINED,
RSN=00F30006**

Explanation:

The DB2 subsystem is not defined to z/OS. Processing terminates.

User response:

Verify that the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ25523I CONNECTED TO DB2 SUBSYSTEM:
ssid VERSION: *nnn***

Explanation:

A connection has been established to the DB2 subsystem. Processing continues.

User response:

No action is required.

CKZ25535E DB2 STOPPED PREMATURELY

Explanation:

The DB2 subsystem stopped prematurely. The DB2 subsystem stopped before it became ready. Processing terminates.

User response:

Determine why the DB2 subsystem did not complete its start up.

**CKZ25536E *name* DID NOT START WITHIN
keyword TIME LIMIT**

Explanation:

The DB2 subsystem did not complete its start up within the time limit specified in the MSTR-DETECT-WAIT or WAIT keyword. For the MSTR-DETECT-WAIT keyword, the xxxMSTR address space did not start executing on z/OS. The xxxMSTR address space may have experienced a JCL error. For the WAIT keyword, the xxxMSTR address space is executing on z/OS, but

DB2 did not become ready for work. Processing terminates.

User response:

Determine why the DB2 subsystem did not complete its start up. If the wait time limit is too short, increase it.

CKZ25537I DB2 HAS STARTED | DB2 HAS TERMINATED

Explanation:

The DB2 subsystem has started or terminated. Processing continues.

User response:

No action is required.

CKZ25538I WAITING FOR DB2 TO START | WAITING FOR DB2 TO TERMINATE

Explanation:

Db2 Cloning Tool is waiting for the DB2 subsystem to start or terminate. Processing continues.

User response:

No action is required.

CKZ25539E DB2 IS ALREADY RUNNING

Explanation:

The DB2 subsystem is already running. Processing terminates.

User response:

Determine why the DB2 subsystem is running.

CKZ25540I START COMMAND: command

Explanation:

Displays the start command that will be issued. Processing continues.

User response:

No action is required.

CKZ25541I COMMAND NOT EXECUTED DUE TO SIMULATION MODE

Explanation:

The start command was not issued because this is a simulation run. Processing continues.

User response:

No action is required.

CKZ25542E UNABLE TO DETERMINE DB2 STATUS; DB2 IS RUNNING BUT IS NOT ACCEPTING CONNECTIONS

Explanation:

Db2 Cloning Tool is unable to determine the status of the DB2 subsystem. The DB2 subsystem is running but is not accepting connections. Processing terminates.

User response:

Get the DB2 subsystem either all the way up or stop it.

CKZ25543I RESTART REPLY: reply

Explanation:

Displays the restart reply that will be issued.

User response:

No action is required.

CKZ25544I REPLY NOT EXECUTED DUE TO SIMULATION MODE

Explanation:

The restart reply was not issued because this is a simulation run. Processing continues.

User response:

No action is required.

CKZ25545E macro ERROR; FUNCTION=function RC=nnnn RSN=X'nnnnnnnn'

Explanation:

An error occurred calling a system macro. Function is the macro function requested, RC is the return code, and RSN is the reason. Processing terminates.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25546E INTERNAL ERROR; LOC=lllll reason text

Explanation:

An internal error has occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ25548I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how DB2START will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ25550E ERROR IN PARAMETERS FOR keyword

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ25551E **REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ25553E **KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ25554E **KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ25556E **NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ25558E **INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ25560E **section NOT AS EXPECTED**

Explanation:

The section specified does not have the expected format. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ25561E **SPECIAL DSNZPARM DOES NOT
HAVE DB2 CATALOG UPDATABLE
ATTRIBUTE**

Explanation:

The special dsnzparm does not have the DB2 catalog updatable attribute. Processing terminates.

User response:

Correct the special dsnzparm to have the DB2 catalog updatable attribute.

CKZ25562E **SPECIAL DSNZPARM DOES NOT
HAVE DEFER,ALL ATTRIBUTE**

Explanation:

The special dsnzparm does not have the DEFER,ALL attribute. Processing terminates.

User response:

Correct the special dsnzparm to have the DEFER,ALL attribute.

CKZ25563E **UNABLE TO LOAD SPECIAL
DSNZPARM MODULE**

Explanation:

The specified dsnzparm module was not able to be loaded. Processing terminates.

User response:

Verify the //STEPLIB includes the load library the contains the special dsnzparm.

CKZ25564E **DSNZPARM MUST BE SPECIFIED
WHEN SPECIAL IS SPECIFIED**

Explanation:

The SPECIAL keyword was specified but the DSNZPARM keyword was not specified. Processing terminates.

User response:

Add the DSNZPARM keyword with the name of the special dsnzparm module.

CKZ25565I **WAIT TIME IS LESS THAN MSTR-
DETECT-WAIT TIME - WAIT
TIMEOUT WILL HAPPEN BEFORE
MSTR-DETECT-WAIT TIMEOUT**

Explanation:

The time specified in the WAIT keyword is less than the time specified in the MSTR-DETECT-WAIT keyword. This will cause the MSTR-DETECT-WAIT timeout to be disabled, as the WAIT timeout will happen before the MSTR-DETECT-WAIT timeout can happen.

User response:

No action is required. If desired, the MSTR-DETECT-WAIT time can be changed to be less than the WAIT time.

CKZ25570I **STARTING DB2 SUBSYSTEM: ssid**

Explanation:

The DB2 subsystem is being started. Processing continues.

User response:

No action is required.

CKZ25586I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

**CKZ26001I hh:mm:ss DB2XCFCLEAN
STARTED - PROGRAM REV=rrr |
hh:mm:ss DB2XCFCLEAN
COMPLETED; RETURN CODE=nnn**

Explanation:

DB2XCFCLEAN processing message.

User response:

No action is required.

**CKZ26003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ26004E DDNAME MISSING: ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

**CKZ26005E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ26005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ26008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ26009E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ26011E JOURNAL DB2 CONTROL RECORD
NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

**CKZ26012E JOURNAL DB2 CONTROL RECORD
IS WRONG VERSION**

Explanation:

The journal DB2 control record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ26020E NO DB2 GROUP NAME

Explanation:

A DB2 group name was not found in the journal DB2 control record.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ26048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how DB2XCFCLEAN will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ26051E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ26053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ26054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ26056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ26058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ26501I hh:mm:ss DB2STOP STARTED - PROGRAM REV=rrr | hh:mm:ss DB2STOP COMPLETED; RETURN CODE=nnn

Explanation:

DB2STOP processing message.

User response:

No action is required.

CKZ26507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ26508E UNABLE TO LOAD PROGRAM: program name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ26520E DSNALI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation:

An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ26521E IFI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation:

An error occurred using the DB2 IFI interface. Function is the IFI function requested and RSN is the DB2 error reason. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ26522E DB2 SUBSYSTEM: db2ssid NOT ACTIVE, RSN=00F30002 | DB2 SUBSYSTEM: db2ssid NOT DEFINED, RSN=00F30006

Explanation:

The DB2 subsystem is not active or not defined to z/OS. Processing terminates.

User response:

Verify that the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ26523I CONNECTED TO DB2 SUBSYSTEM: ssid VERSION: nnn

Explanation:

A connection has been established to the DB2 subsystem. Processing continues.

User response:

No action is required.

CKZ26524W IFI WARNING; FUNCTION=function RC=return_code RSN=reason_code

Explanation:

A warning occurred using the Db2 IFI interface. *function* is the IFI function requested and *reason_code* is the Db2 error reason. Processing terminates.

User response:

No action is required.

CKZ26525I CANNOT DETERMINE IRLMPROC NAME. THE DEFAULT NAME name WILL BE USED

Explanation:

If the -DIS GROUP command ends with return code 4 or higher, the name of the IRLM procedure cannot be retrieved from the result. In this case, the generated default value is used.

User response:

No action is required.

CKZ26534E USERID IS NOT AUTHORIZED TO ISSUE THE DB2 COMMAND

Explanation:

The userid running the job is not authorized to issue the DB2 command. Processing terminates.

User response:

Verify the userid running the job has SYSADM authority.

CKZ26535E COMMAND FAILED; RC= nnnn

Explanation:

The DB2 command failed. Processing terminates.

User response:

Check the messages from the command failure and correct the problem. If unable to resolve the problem, contact IBM Software Support.

CKZ26536E DB2 DID NOT TERMINATE WITHIN WAIT TIME LIMIT

Explanation:

The DB2 subsystem did not complete its termination within the time limit specified in the WAIT keyword. Processing terminates.

User response:

Determine why the DB2 subsystem did not complete its termination. If the wait time limit is too short, increase it.

CKZ26537I DB2 HAS TERMINATED

Explanation:

The DB2 subsystem has terminated. Processing continues.

User response:

No action is required.

CKZ26538I WAITING FOR DB2 TO TERMINATE

Explanation:

Db2 Cloning Tool is waiting for the DB2 subsystem to terminate. Processing continues.

User response:

No action is required.

CKZ26539I name IS STILL RUNNING

Explanation:

The indicated DB2 address space is still running. Processing continues.

User response:

No action is required.

CKZ26540I DB2 COMMAND: command

Explanation:

Displays the DB2 command that will be issued. Processing continues.

User response:

No action is required.

CKZ26541I COMMAND NOT EXECUTED DUE TO SIMULATION MODE

Explanation:

The DB2 command was not issued because this is a simulation run. Processing continues.

User response:

No action is required.

CKZ26542E UNABLE TO DETERMINE DB2 STATUS; DB2 IS RUNNING BUT IS NOT ACCEPTING CONNECTIONS

Explanation:

Db2 Cloning Tool is unable to determine the status of the DB2 subsystem. The DB2 subsystem is running but is not accepting connections. Processing terminates.

User response:

Get the DB2 subsystem either all the way up or stop it.

CKZ26543I DB2 SUBSYSTEM TERMINATION NOT COMPLETE

Explanation:

The DB2 subsystem has not completely terminated. Processing continues.

User response:

No action is required.

CKZ26548I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how DB2STOP will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ26550E ERROR IN PARAMETERS FOR keyword

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ26551E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ26553E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ26554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ26556E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ26558E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ26560E section NOT AS EXPECTED

Explanation:

The section specified does not have the expected format. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ26570I STOPPING DB2 SUBSYSTEM: ssid

Explanation:

The DB2 subsystem is being stopped. Processing continues.

User response:

No action is required.

CKZ26586I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

CKZ27001I hh:mm:ss DB2ALTERBDS STARTED - PROGRAM REV=rrr | hh:mm:ss DB2ALTERBDS COMPLETED; RETURN CODE=nnn

Explanation:

DB2ALTERBDS processing message.

User response:

No action is required.

CKZ27003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ27004E DDNAME MISSING: ddname**Explanation:**

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ27005E ALLOCATION FAILED FOR DSN: datasetname**Explanation:**

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ27005W DEALLOCATION FAILED FOR DDNAME: ddname**Explanation:**

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ27007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn**Explanation:**

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ27008E UNABLE TO LOAD PROGRAM: program name**Explanation:**

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ27009E ERROR ACCESSING JOURNAL FILE; LOC=lllll**Explanation:**

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ27011E JOURNAL type RECORD NOT FOUND**Explanation:**

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ27012E JOURNAL type RECORD IS WRONG VERSION**Explanation:**

A journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. A journal created by a prior release of Db2 Cloning Tool can be upgraded to the current release by using the Db2 Cloning Tool JRNLUPEGRADE command. See the "JRNLUPEGRADE" on page 470 topic for the scenarios where JRNLUPEGRADE can be used. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ27013E COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc LOC=lllll**Explanation:**

There was a problem with the journal records needed to initiate the command. The number of records read from the journal, rrrr, is not the same as the number indicated in the journal DB2 control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ27020E NOTHING WAS REQUESTED TO BE DONE

Explanation:

No request keywords were included in the DB2ALTERBSDS command. Processing terminates.

User response:

Add a request keyword to the DB2ALTERBSDS command.

CKZ27030I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options
Explanation:

Informational message indicating how DB2ALTERBSDS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ27040E DB2UPDATE HAS NOT BEEN RUN FOR THIS MEMBER
Explanation:

DB2UPDATE has not been run for this member. Processing terminates.

User response:

Run DB2UPDATE for the member.

CKZ27041E DB2UPDATE FOR THIS MEMBER WAS SIMULATION, SIMULATION MUST BE SPECIFIED
Explanation:

DB2UPDATE for this member was run as a simulation and this run is not a simulation. Processing terminates.

User response:

Run as SIMULATE or run DB2UPDATE for this member without SIMULATE.

CKZ27042E SLB-START WAS REQUESTED BUT THERE WAS NO SLB INFORMATION FOUND IN A BSDS
Explanation:

SLB-START was requested, but no SLB information was found in a BSDS. Processing terminates.

User response:

Verify that the BSDS should have SLB information in it. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ27050E ERROR IN PARAMETERS FOR keyword
Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ27051E REQUIRED KEYWORD MISSING: keyword
Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ27052E REQUIRED KEYWORD NOT ALLOWED: keyword FOR NON DATA SHARING DB2
Explanation:

The DB2 system being updated is not part of a data group and a keyword that applies only to data sharing groups has been specified. Processing terminates.

User response:

Remove the not allowed keyword from the command.

CKZ27053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED
Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ27054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword
Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ27056E NOTHING SPECIFIED FOR KEYWORD: keyword
Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ27058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text
Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ27086I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

**CKZ27101I hh:mm:ss BSDS ALTER STARTED -
PROGRAM REV=rrr | hh:mm:ss
BSDS ALTER COMPLETED;
RETURN CODE=nnn**

Explanation:

BSDS ALTER processing message.

User response:

No action is required.

**CKZ27103I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ27104E OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27105E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ27105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ27106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ27107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ27109E ERROR ACCESSING BSDS FILE;
LOC=lllll**

Explanation:

A VSAM error occurred accessing a file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ27110E ERROR ACCESSING ACTIVE LOG;
DDNAME: ddname R15=nnnn
ERROR=nnn FUNCTION=function
LOC=lllll RBA OF RECORD:
X'nnnnnnnn_nnnnnnn'**

Explanation:

A VSAM error occurred accessing the indicated active log file. Processing terminates.

User response:

If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27120E BSDS CONTROL RECORD NOT
FOUND**

Explanation:

The control record was not found in the BSDS. The BSDS is not valid. Processing terminates.

User response:

Determine why the BSDS became invalid. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27121I HIGHEST LOG RBA
WRITTEN=xxxxxxxxxxxx**

Explanation:
Informational message showing the highest log rba written recorded in the BSDS.

User response:
No action is required.

**CKZ27122E BSDS RECORD NOT FOUND FOR
recordtype**

Explanation:
An expected BSDS record type was not found. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

**CKZ27123I ACTIVE LOG COPY n
DSN=datasetname START
RBA=xxx**

Explanation:
This is an informational message that displays the data set name and start RBA of an active log.

User response:
No action is required.

**CKZ27124I | ACTIVE LOG DOES NOT CONTAIN
CKZ27124W SLB LRSN; CONDITIONAL
RESTART RECORD WILL NOT BE
CREATED**

Explanation:
The active log of this Db2 subsystem does not contain the SLB LRSN being used for a conditional restart. A conditional restart record will not be created for this Db2 subsystem. If SLB-START-NOT-CREATED(RC(nn)) is specified with a non-zero value, then CKZ27124W is issued; otherwise, CKZ27124I is issued.

User response:
No action is required.

**CKZ27135E DDNAME=ddname IS ALREADY
ALLOCATED**

Explanation:
The ddname is already allocated and it must not be.

User response:
Ensure the ddname is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27140E DSNJU003 UTILITY EXECUTION
FAILED**

Explanation:
The invoked DSNJU003 utility has failed.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ27150W NO ARCHIVE LOGS TO DELETE

Explanation:
No archive logs met the selection criteria for deletion from the BSDS.

User response:
No action is required.

CKZ27151W NO ACTIVE LOGS TO DELETE

Explanation:
No active logs met the selection criteria for deletion from the BSDS.

User response:
No action is required.

**CKZ27152E SLB ENDLRSN IS OUT OF LAST
CHECKPOINT**

Explanation:
The SLB LRSN is less than LRSN of the last checkpoint in BSDS, but NOT-DB2-SLB(RC) is specified with a value not less than 8. The DB2ALTERBSDS job is terminated.

User response:
If you want to continue the DB2ALTERBSDS job when the SLB LRSN is less than the LRSN of the last checkpoint in the BSDS, specify a RC value of less than 8 and resubmit the job.

**CKZ27152I SLB ENDLRSN IS OUT OF LAST
CHECKPOINT**

Explanation:
The SLB LRSN is less than LRSN of the last checkpoint in BSDS, but NOT-DB2-SLB(RC) is specified with a value of 0.

User response:
No action is required.

**CKZ27152W SLB ENDLRSN IS OUT OF LAST
CHECKPOINT**

Explanation:
The SLB LRSN is less than LRSN of the last checkpoint in BSDS, but NOT-DB2-SLB(RC) is specified with a value greater than 0 but less than 8.

User response:
No action is required.

**CKZ27153E LAST CHECKPOINT HAS
SHUTDOWN STATUS**

Explanation:

The LRSN of the last checkpoint has the status "shutdown checkpoint," and NOT-DB2-SLB(RC) is specified with a value not less than 8. The DB2ALTERBSDS job is terminated.

User response:

To continue the DB2ALTERBSDS job when the LRSN of the last checkpoint has the status "shutdown checkpoint," specify a RC value of less than 8 and resubmit the job.

CKZ27153I LAST CHECKPOINT HAS SHUTDOWN STATUS

Explanation:

The LRSN of the last checkpoint has the status "shutdown checkpoint," and NOT-DB2-SLB(RC) is specified with a value of 0.

User response:

No action is required.

CKZ27153W LAST CHECKPOINT HAS SHUTDOWN STATUS

Explanation:

The LRSN of the last checkpoint has the status "shutdown checkpoint," and NOT-DB2-SLB(RC) is specified with a value greater than 0 but less than 8.

User response:

No action is required.

CKZ27154I CONDITIONAL RESTART RECORD IS ADDED TO BSDS

Explanation:

The SLB ENDLRSN is less than the LRSN of the last checkpoint in the BSDS, or the LRSN of the last checkpoint has the status "shutdown checkpoint." The NOT-DB2-SLB(RC(*nnnn*), CRCR) was specified with a RC value less than 8. In this case, the conditional restart record will be added to BSDS.

User response:

No action is required.

CKZ27155I CONDITIONAL RESTART RECORD IS NOT ADDED TO BSDS

Explanation:

The SLB ENDLRSN is less than the LRSN of the last checkpoint in the BSDS, or the LRSN of the last checkpoint has the status "shutdown checkpoint." The NOT-DB2-SLB(RC(*nnnn*), NOCRCR) was specified with a RC value of 8 or greater. In this case, the conditional restart record will not be added to BSDS.

User response:

No action is required.

CKZ27501I hh:mm:ss DB2SETLOG STARTED - PROGRAM REV=rrr | hh:mm:ss DB2SETLOG COMPLETED; RETURN CODE=nnn

Explanation:

DB2SETLOG processing message.

User response:

No action is required.

CKZ27507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ27508E UNABLE TO LOAD PROGRAM: program name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ27520E DSNALI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation:

An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ27521E IFI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation:

An error occurred using the DB2 IFI interface. Function is the IFI function requested and RSN is the DB2 error reason. Processing terminates.

User response:

Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ27522E DB2 SUBSYSTEM: db2ssid NOT ACTIVE, RSN=00F30002 | DB2 SUBSYSTEM: db2ssid NOT DEFINED, RSN=00F30006

Explanation:

The DB2 subsystem is not active or not defined to z/OS. Processing terminates.

User response:

Verify that the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ27523I **CONNECTED TO DB2 SUBSYSTEM:
ssid VERSION: nnn**

Explanation:

A connection has been established to the DB2 subsystem. Processing continues.

User response:

No action is required.

CKZ27534E **USERID IS NOT AUTHORIZED TO
ISSUE THE DB2 COMMAND**

Explanation:

The userid running the job is not authorized to issue the DB2 command. Processing terminates.

User response:

Verify the userid running the job has SYSADM authority.

CKZ27535E **COMMAND FAILED; RC= nnnn**

Explanation:

The DB2 command failed. Processing terminates.

User response:

Check the messages from the command failure and correct the problem. If unable to resolve the problem, contact IBM Software Support.

CKZ27540I **DB2 COMMAND: command**

Explanation:

Displays the DB2 command that will be issued. Processing continues.

User response:

No action is required.

CKZ27541I **COMMAND NOT EXECUTED DUE
TO SIMULATION MODE**

Explanation:

The start command was not issued because this is a simulation run. Processing continues.

User response:

No action is required.

CKZ27548I **OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation:

Informational message indicating how DB2SETOG will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ27551E **REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ27553E **KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ27554E **KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ27556E **NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ27570I **SUSPENDING DB2 SUBSYSTEM:
ssid | RESUMING DB2
SUBSYSTEM: ssid**

Explanation:

The DB2 subsystem is being suspended or resumed. Processing continues.

User response:

No action is required.

CKZ28001I **hh:mm:ss DB2SCHEMA-UPDATE
STARTED - PROGRAM REV=
revision | hh:mm:ss DB2SCHEMA-
UPDATE COMPLETED; RETURN
CODE= return_code**

Explanation:

This message provides DB2SCHEMA-UPDATE processing information.

User response:

No action is required.

**CKZ28002E SCHEMA-MASKS WRONG VALUE
 SPECIFIED: *value***

Explanation:

The value in the keyword is invalid. The specified mask does not meet the requirements. Processing terminates.

User response:

Correct the value that is specified in the keyword.

**CKZ28003I DDNAME= *ddname* ALLOCATED
 FOR DSN=*datasetname***

Explanation:

The ddname that is listed in the message has been successfully allocated for the indicated data set.

User response:

No action is required.

CKZ28004E DDNAME MISSING: *ddname*

Explanation:

The ddname that is listed in the message was specified for Db2 Cloning Tool to use, but the ddname was not found. Processing terminates.

User response:

Either correct the specified ddname, or add the appropriate ddname to the job's JCL.

**CKZ28005E ALLOCATION FAILED FOR DSN:
 *datasetname***

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ28005W DEALLOCATION FAILED FOR
 DDNAME: *ddname***

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ28006E ERROR CALLING CKZ01VV1
 internal_table FUNCTION: *function*
 R15=*nnnn* R0=*nnnnnnnn*
 LOC=*location_of_error***

Explanation:

An error occurred while using a data space. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing these messages and the PARMLIB member that controls execution of Db2 Cloning Tool.

**CKZ28007W ERROR CALLING CKZ01HEX;
 FUNCTION=*function* R15=*nnnn***

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

**CKZ28008E UNABLE TO LOAD PROGRAM:
 *program_name***

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Ensure that the job's STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ28009E ERROR ACCESSING JOURNAL
 FILE; LOC=*location***

Explanation:

A VSAM error occurred while accessing a file. Processing terminates.

User response:

See the associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ28010E DUPLICATE JOURNAL ENTRY;
 LOC=*location***

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains these messages.

CKZ28011E JOURNAL *record_type* NOT FOUND

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. The *record_type* that is listed in the message is the type of record that is not found. Processing terminates.

User response:

Verify that the same value was used for the DB2-NAME keyword that was used with the prior DB2SQL

command for this DB2 subsystem or data sharing group. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ28012E JOURNAL *record_type* IS WRONG VERSION

Explanation:

The journal record does not match the expected format. The incorrect *record_type* is listed in the message. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve this problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ28013E SCHEMA-MASKS ENTRIES MISMATCH; PREVIOUS RUN HAD DIFFERENT SCHEMA-MASKS ENTRIES

Explanation:

The entries entered for this member in the keyword SCHEMA-MASKS do not match the entries specified for the previous run of DB2SCHEMA-UPDATE. The entries for the previous run are printed. Processing terminates.

User response:

Correct the entries to match those specified for previous run.

CKZ28014E COUNT MISMATCH; RECORDS READ= *number_read* DB2SCHEMA-UPDATE RECORD COUNT= *number_in_control_journal* LOC = *location*

Explanation:

There was a problem with the journal records that are needed to initiate the command. The number of records read from the journal, *number_read*, is not the same as the number indicated in the journal DB2 control record, *number_in_control_journal*. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ28019E SIMULATE AND RESUME ARE MUTUALLY EXCLUSIVE

Explanation:

Mutually exclusive keywords might have been specified. Processing terminates.

User response:

Correct the input parameters.

CKZ28020E DSNALI ERROR; FUNCTION=*function* RC=*return_code* RSN=*reason_code* LOC=*location*

Explanation:

An error occurred calling DSNALI. *function* is the CAF function that was requested and *reason_code* is the DB2 error reason. Processing terminates.

User response:

Check the DB2 messages and codes documentation for the *reason_code* value. If unable to resolve the problem, contact IBM Software Support.

CKZ28022E DB2 SUBSYSTEM: *db2ssid* NOT ACTIVE, RSN=00F30002 | DB2 SUBSYSTEM: *db2ssid* NOT DEFINED, RSN=00F30006

Explanation:

The DB2 subsystem is not active or not defined. Processing terminates.

User response:

Verify that the DB2 subsystem is active and the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ28023I CONNECTED TO DB2 SUBSYSTEM: *ssid* RELEASE: *nnn*

Explanation:

A connection has been established to the DB2 subsystem. Processing continues.

User response:

No action is required.

CKZ28024E PLAN: *planname* NOT USABLE OR MAY NOT EXIST, RSN=00F30040

Explanation:

A bind for this plan has not been done or the plan is not usable. Processing terminates.

User response:

Bind the Db2 Cloning Tool plan to the DB2 subsystem.

CKZ28025I DISCONNECTED FROM DB2 SUBSYSTEM: *ssid*

Explanation:

A connection has been closed to the DB2 subsystem. Processing continues.

User response:

No action is required.

CKZ28026E RESUME NOT SPECIFIED AND PREVIOUS RUN WAS NOT A SIMULATION

Explanation:

A DB2SCHEMA-UPDATE was attempted without the RESUME parameter, but the journal indicates that a non-simulation DB2SCHEMA-UPDATE has already been attempted.

User response:

Specify RESUME for the DB2SCHEMA-UPDATE command.

CKZ28027E RESUME WAS SPECIFIED AND IT IS FIRST RUN THAT IS NOT A SIMULATION

Explanation:

A DB2SCHEMA-UPDATE was attempted with the RESUME parameter, but the journal indicates that DB2SCHEMA-UPDATE has not been run previously as non-simulation.

User response:

Run DB2SCHEMA-UPDATE command without the RESUME parameter.

CKZ28028E KEYWORD *keyword* VALUE WAS CHANGED SINCE PREVIOUS RUN. KEYWORDS VALUES SHOULD NOT BE CHANGED IN CASE OF USING RESUME KEYWORD

Explanation:

A DB2SCHEMA-UPDATE was attempted with the RESUME parameter, but the journal indicates that the DB2SCHEMA-UPDATE keyword value is not the same as it was for previous run.

User response:

Correct the input parameters to match those specified for previous run.

CKZ28030I OPTIONS IN EFFECT FOR THIS EXECUTION: *merged_options*

Explanation:

This informational message indicates how DB2SCHEMA-UPDATE will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ28040E CKZ00900 UNEXPECTED RESULTS; *error_text*

Explanation:

An unexpected condition occurred calling program CKZ00900. *error_text* contains a description of the problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ28051E REQUIRED KEYWORD MISSING: *keywords*

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ28052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=*section* TOKEN=*token* | REQUIRED INI VALUE MISSING FOR SECTION=*section* TOKEN=*token* | INVALID INI VALUE FOR SECTION=*section* TOKEN=*token*

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

CKZ28053E KEYWORD: *keyword* MAXIMUM LENGTH: *length* EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length that is allowed for the operand. *length* is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ28054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: *keyword*

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ28056E NOTHING SPECIFIED FOR KEYWORD: *keyword*

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ28057E DUPLICATE FOUND; KEYWORD: *keyword* ENTRY: *entry*

Explanation:

The indicated *entry* for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ28058E **INVALID VALUE IN KEYWORD:**
keyword **VALUE:** *value error_text*

Explanation:

The value in the keyword is invalid. *error_text* indicates the problem that was detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ28075E **UNMATCHED ENTRIES IN
KEYWORD:** *keyword*

Explanation:

For SCHEMA-MASKS, there must be pairs of source and target masks. An uneven number of entries was specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ28076E **NO ENTRIES FOUND FOR
KEYWORD:** *keyword*

Explanation:

For SCHEMA-MASKS, there must be pairs of source and target masks. No entries were specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ28085I **PAIRS FOR KEYWORD:** *keyword*

Explanation:

Parsing found the listed pairs for the keyword.

User response:

No action is required.

CKZ28101I *hh:mm:ss* **SQL PROCESSOR
STARTED - PROGRAM
REV=*revision* | *hh:mm:ss* SQL
PROCESSOR COMPLETED;
RETURN CODE=*return_code***

Explanation:

This informational message is an SQL PROCESSOR processing message.

User response:

No action is required.

CKZ28106E **ERROR CALLING CKZ01VV1
internal_table FUNCTION:** *function*
**R15=*nnnn* R0=*nnnnnnnn*
LOC=*location***

Explanation:

A problem occurred using a dataspace. *internal_table* is the name of the internal table. *location* is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the PARMLIB member that controls execution of Db2 Cloning Tool.

CKZ28107W **ERROR CALLING CKZ01HEX;
FUNCTION=*function* R15=*nnnn***

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

CKZ28110E **NO SCHEMAS WERE FOUND**

Explanation:

The specified masks do not identify any schema values. Processing terminates.

User response:

Correct the SCHEMA-MASKS entries.

CKZ28120W **DSNTIAR FAILED; RC: *return_code*
RSN: *reason_code***

Explanation:

An error occurred using DSNTIAR to format error messages. Processing continues.

User response:

Report this message to IBM Software Support.

CKZ28121E **SQL *function* FAILED, SQLCODE:
sql_code SQLSTATE: *sql_state***

Explanation:

An error occurred processing an SQL statement. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ28122W **MASK IDENTIFIES A *field* OF
TABLE ON WHICH COLUMN
MASKS OR ROW PERMISSIONS
DEPENDS. MASK, TABLE CREATOR
AND NAME FOLLOW | MASK
IDENTIFIES A *field* OF INDEX
WHICH WAS CREATED USING
SCALAR EXPRESSION. MASK,
INDEX CREATOR AND NAME
FOLLOW**

Explanation:

The specified mask identifies a *field* value for table or index that cannot be processed by CATMAINT.

User response:

If possible, delete and re-create problematic objects after the DB2SCHEMA-UPDATE step. If re-creation of

objects is not an option, change SCHEMA-MASKS entries.

CKZ28124E THERE ARE OBJECTS THAT CANNOT BE PROCESSED BY DB2 CATMAINT

Explanation:

The specified masks identify objects that cannot be processed. Processing terminates.

User response:

If possible, delete and re-create problematic objects after the DB2SCHEMA-UPDATE step. If re-creation of objects is not an option, change SCHEMA-MASKS entries.

CKZ28125W MORE THAN ONE OLD SCHEMA VALUE WILL BE TRANSLATED INTO FOLLOWING NEW SCHEMA VALUE: *value*

Explanation:

The SCHEMA-MASKS entries specify one new schema *value* for more than one source schema values.

User response:

Correct the SCHEMA-MASKS entries.

CKZ28126E THERE ARE DUPLICATES IN (OLD SCHEMA VALUE, NEW SCHEMA VALUE) PAIRS, WHICH CANNOT BE PROCESSED BY DB2 CATMAINT. SCHEMA PAIRS SHOULD NOT CONTAIN DUPLICATES, SPECIFIED MASKS SHOULD NOT DEFINE ONE NEW SCHEMA VALUE FOR MORE THAN ONE OLD SCHEMA VALUE

Explanation:

The SCHEMA-MASKS entries specify one new schema *value* for more than one source schema values.

User response:

Correct the SCHEMA-MASKS entries.

CKZ28127W NEW SCHEMA VALUE *value* IS ALREADY PRESENT ON TARGET SSID

Explanation:

DB2 CATMAINT cannot process (old schema value, new schema value) pairs if the new schema value is already present on the target SSID.

User response:

Correct the SCHEMA-MASKS entries.

CKZ28128E THERE ARE NEW SCHEMA VALUES THAT ARE ALREADY PRESENT ON TARGET SSID, WHICH CANNOT BE PROCESSED BY DB2 CATMAINT. SPECIFIED MASKS SHOULD NOT

DEFINE NEW SCHEMA VALUES THAT ALREADY EXIST ON TARGET SSID

Explanation:

DB2 CATMAINT cannot process (old schema value, new schema value) pairs if the new schema value is already present on the target SSID.

User response:

Correct the SCHEMA-MASKS entries.

CKZ28201I *hh:mm:ss* CATMAINT PROCESSOR STARTED - PROGRAM REV=*revision* | *hh:mm:ss* CATMAINT PROCESSOR COMPLETED; RETURN CODE=*return_code*

Explanation:

This informational message is a CATMAINT PROCESSOR processing message.

User response:

No action is required.

CKZ28203I DDNAME=*ddname* ALLOCATED FOR DSN=*data_set_name*

Explanation:

ddname has been successfully allocated for the indicated data set.

User response:

No action is required.

CKZ28204E OPEN FAILED FOR DDNAME=*ddname*

Explanation:

ddname was allocated for Db2 Cloning Tool to use, but the open for the file failed. Processing terminates.

User response:

If unable to determine the reason that the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ28205E ALLOCATION FAILED FOR DSN: *data_set_name*

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains this message.

CKZ28205W DEALLOCATION FAILED FOR DDNAME: *ddname*

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains this message.

**CKZ28207W ERROR CALLING CKZ01HEX;
FUNCTION=function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

**CKZ28235E DDNAME=ddname IS ALREADY
ALLOCATED**

Explanation:

The ddname is already allocated. The ddname must not be allocated.

User response:

Ensure that the ddname is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

**CKZ28236E LOAD FOR CKZCATM MODULE
FAILED**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ28240E CKZCATM EXECUTION FAILED

Explanation:

An error occurred during CKZCATM execution.

User response:

Check the CKZCATM output to identify the possible cause of an error. If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ28241E DSNUTILB UTILITY EXECUTION
FAILED**

Explanation:

The invoked utility failed.

User response:

If unable to determine the reason for the failure from the associated messages, contact IBM Software

Support. Have available the listing containing these messages.

**CKZ28260E ATTACH ERROR; RETURN CODE=
return_code REASON CODE=
reason_code**

Explanation:

An attempt to attach a subtask failed with the return code and reason code that are listed in the message.

User response:

Ensure that the DB2 runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

**CKZ28301I hh:mm:ss STATEMENTS
PROCESSOR STARTED - PROGRAM
REV=revision | STATEMENTS
PROCESSOR COMPLETED;
RETURN CODE=return_code**

Explanation:

This informational message is a STATEMENT PROCESSOR processing message.

User response:

No action is required.

**CKZ28302E OPEN FAILED FOR DDNAME:
ddname**

Explanation:

Open has failed for *ddname*. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ28303E DATA SET SPECIFIED FOR
DDNAME ddname DOES NOT HAVE
A LRECL OF 80**

Explanation:

The data set that is allocated to the ddname does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response:

Change the data set that is allocated to the ddname to have a LRECL of 80.

**CKZ28304I NUMBER OF type_of_statement
STATEMENTS PROCESSED: nnn**

Explanation:

This informational message lists the number of successfully executed DROP or CREATE statements.

User response:

No action is required.

**CKZ28305E INVALID TERMINATOR
SPECIFIED: character_value**

Explanation:

An invalid character value was found for an SQL terminator. A blank, comma, double quotation mark, left parenthesis, right parenthesis, single quotation mark, or underscore cannot be used as an SQL terminator.

User response:

If it was specified manually, correct the character value for the SQL terminator. If the character value was not specified manually, contact IBM Software Support. Have available the listing that contains this message.

CKZ28306E ERROR CALLING CKZ01VV1
internal_table* FUNCTION: *function
R15=*nnnn* R0=*nnnnnnnn*
LOC=*location*

Explanation:

A problem occurred using a dataspace. *internal_table* is the name of the internal table and *location* is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ28307W ERROR CALLING CKZ01HEX;
FUNCTION=*function* R15=*nnnn*

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Report this message to IBM Software Support.

CKZ28308E STATEMENT LENGTH EXCEEDED

Explanation:

An SQL statement is too long. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ28320W DSNTIAR FAILED; RC: *return_code*
RSN: *reason_code*

Explanation:

An error occurred when using DSNTIAR to format error messages. Processing continues.

User response:

Report this message to IBM Software Support.

CKZ28321E SQL *function* FAILED, SQLCODE:
sql_code* SQLSTATE: *sql_state

Explanation:

An error occurred processing an SQL statement. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ40001I hh:mm:ss BCSCLEAN STARTED -
PROGRAM REV=*rrr* | hh:mm:ss
BCSCLEAN COMPLETED; RETURN
CODE=*nnn*

Explanation:

BCSCLEAN command processing message.

User response:

No action is required.

CKZ40003I DDNAME=*ddname* ALLOCATED
FOR DSN=*datasetname*

Explanation:

'*ddname*' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ40004E DDNAME MISSING: *ddname*

Explanation:

'*ddname*' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the *ddname* specified, or, add the appropriate *ddname* to the job's JCL.

CKZ40005E ALLOCATION FAILED FOR DSN:
datasetname

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ40005W DEALLOCATION FAILED FOR
DDNAME: *ddname*

Explanation:

Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ40008E UNABLE TO LOAD PROGRAM:
 program name**

Explanation:

The indicated program name was not found.
Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ40009E ERROR ACCESSING JOURNAL
 FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file.
Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ40010E No dsnmask specified for
 CLEANUP-CATALOG-DSNMASKS –
 at least 1 dsnmask is required.**

Explanation:

The CLEANUP-CATALOG-DSNMASKS keyword requires at least one data set name mask.

User response:

Supply at least one data set name mask for the keyword.

**CKZ40011E JOURNAL CONTROL RECORD NOT
 FOUND | JOURNAL CONTROL
 RECORD IS WRONG VERSION |
 JOURNAL TARGET CATALOG
 RECORD(S) NOT FOUND |
 JOURNAL TARGET CATALOG
 RECORD IS WRONG VERSION**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ40013E COUNT MISMATCH; RECORDS
 READ=*number_read* CONTROL
 RECORD COUNT=*count*
 LOC=*location***

Explanation:

There was a problem with the journal records that are needed to initiate the command. The number of records read from the journal (*number_read*) is not the same as the number that is indicated in the journal DB2 control record (*count*). Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ40048I OPTIONS IN EFFECT FOR THIS
 EXECUTION: merged options**

Explanation:

Informational message indicating how BCSCLEAN will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

**CKZ40051E REQUIRED KEYWORD MISSING:
 keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ40052E REQUIRED INI SECTION/TOKEN
 MISSING: SECTION=*section*
 TOKEN=*token* | REQUIRED INI
 VALUE MISSING FOR
 SECTION=*section* TOKEN=*token* |
 INVALID INI VALUE FOR
 SECTION=*section* TOKEN=*token***

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

**CKZ40053E KEYWORD: *keyword* MAXIMUM
 LENGTH: *nnn* EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. *nnn* is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ40054E KEYWORD HAS MORE THAN 1
 OPERAND; ONLY 1 ALLOWED:
 keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ40055E Dsname mask has invalid characters at offset: *offset* mask: *data_set_name_mask*.

Explanation:

Validation of the data set name mask failed. The error offset (starting with 0) is given in the message.

User response:

Review the data set name masking requirements in the documentation, and make the necessary changes.

CKZ40056E **NOTHING SPECIFIED FOR KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ40057E Dsname mask too long: *length* *data_set_name_mask*.

Explanation:

Data set name masks must be 44 characters or less. The data set name mask in error and its length are given in the message.

User response:

Review the data set name masking requirements in the documentation, and make the necessary changes.

CKZ40058E **INVALID VALUE IN KEYWORD: keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ40059I **TARGET CATALOGS ON TARGET VOLUMES**

Explanation:

There are target catalogs on the target volumes.

User response:

No action is required.

CKZ41001I hh:mm:ss FINDUCATS STARTED - PROGRAM REV=rrr | hh:mm:ss FINDUCATS COMPLETED; RETURN CODE=nnn

Explanation:

FINDUCATS processing message.

User response:

No action is required.

CKZ41006E **ERROR CALLING CKZ01VV1 ttttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. ttttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ41008E **UNABLE TO LOAD PROGRAM: program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ41051E **REQUIRED KEYWORD MISSING: keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ41053E **KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ41054E **KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ41056E **NOTHING SPECIFIED FOR KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ41057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

**CKZ41058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ41060E UCBSCAN ERROR; RETURN
CODE=nn REASON CODE=nn |
UCBINFO ERROR; RETURN
CODE=nn REASON CODE=nn**

Explanation:

An error occurred using UCBSCAN or UCBINFO. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ41061E CKZ01SMF ERROR; RETURN
CODE=nnnn LOC: llllll entry**

Explanation:

An error occurred using CKZ01SMF to obtain SSI information for the 'entry'. llllll is the internal location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ41062E *Multiple messages and
explanations; see Explanation***

Explanation**Message**

NO STORAGE GROUPS RETURNED BY SSI

Explanation

The FINDUCATS command was requested to determine volume serial numbers by storage group names. SSI did not return any storage groups to CKZ01SMF. Processing terminates.

Message

NO VOLUME SERIALS RETURNED BY SSI

Explanation

The FINDUCATS command was requested to check the SMS status of volume serials. SSI did not return any volume serials to CKZ01SMF. Processing terminates.

User response:

Check that storage groups are defined on the system. If unable to determine the cause of this message, contact IBM Software Support. Have available the listing containing this message.

**CKZ41063W NO VOL/STG MATCH FOUND;
KEYWORD: keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was not matched. For VOL, the indicated volser, or, the volser derived from a storage group, was not found. For STG, the indicated storage group was not found. Processing continues.

User response:

No action is required.

**CKZ41064E INVALID VOLSER: volser IN
KEYWORD: keyword**

Explanation:

The volume serial number specified is invalid. Processing terminates.

User response:

Correct the volser specification.

**CKZ41067W NO VOLUME SERIALS FOR
STORAGE GROUP: storage group**

Explanation:

SSI did not return any volsers for the storage group to CKZ01SMF. Processing continues.

User response:

No action is required.

CKZ41075E NO VOLUME SERIALS SELECTED

Explanation:

No volume serials have been found for FINDUCATS to examine. Processing terminates.

User response:

Check if volsers specified in the FROM- keyword have been removed by use of an EXCLUDE-FROM- keyword.

**CKZ41086I STORAGE GROUPS/MASKS FOR
KEYWORD: keyword**

Explanation:

Parsing found the listed storage groups/masks for the keyword.

User response:

No action is required.

CKZ41087I **nnnnn VOLSERS OR MASKS FOR
KEYWORD: keyword list of volsers
| nnnnn VOLSERS DERIVED FOR
KEYWORD: keyword list of volsers
| nnnnn VOLSERS RESOLVED FOR
KEYWORD: keyword**

Explanation:

Parsing found the listed volsers/masks for a keyword, or, derived the listed volsers from a storage group keyword. nnnnn is the number of volume serials. The third format, RESOLVED, indicates the number of volume serials found online for the associated keyword.

User response:

No action is required.

CKZ41101I **hh:mm:ss VOLUME COLLECTION
STARTED - PROGRAM REV=rrr |
hh:mm:ss VOLUME COLLECTION
COMPLETED; RETURN CODE=nnn**

Explanation:

FINDUCATS volume processing message.

User response:

No action is required.

CKZ41103I **DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ41104E **OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was allocated for Db2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response:

If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ41105E **ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ41105W **DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ41106E **ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ41108E **UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ41130E **AMSOPEN FAILED; R15=nnnn**

Explanation:

An attempt was made to issue an IDCAMS command. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ41131E **NON-ZERO RETURNED BY
IDCAMS; RC=nnnn**

Explanation:

An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will terminate for an IDCAMS return code greater than 4.

User response:

Check the volume that caused the DCOLLECT errors. Correct the problems with the volume.

CKZ41135I **hh:mm:ss VOLSER volser
processing step**

Explanation:
FINDUCATS volume processing message.

User response:
No action is required.

CKZ41136I Report by volume serial numbers

Explanation:
Indicates the start of the FINDUCATS report by volume serial number(s).

User response:
No action is required.

CKZ41137I Report of user catalogs on all examined volume serials

Explanation:
Indicates the start of the FINDUCATS report of catalogs.

User response:
No action is required.

CKZ41141E CKZ00900 UNEXPECTED RESULTS; error text

Explanation:
An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ41142E BAD MLA VALUE FOUND: mla value

Explanation:
The MLA value found was invalid for Db2 Cloning Tool processing. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ41143E CSI STORAGE AREA WILL EXCEED 1MB

Explanation:
Program IGGCSI00, Catalog Search Interface, required more storage to locate alias names. The amount of storage will exceed 1,048,575. Processing terminates.

User response:
Contact IBM Software Support. Have available the listing that contains this message.

CKZ41144E CSI RETURNED NO DATA

Explanation:
This should not occur. Program IGGCSI00, Catalog Search Interface, did not return any data. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ41145W NO DATA SETS PROCESSED FROM VOLUME(S)

Explanation:
No data sets other than SYS1.VTOCIX or SYS1.VVDS entries were found on the specified volumes. Processing continues.

User response:
No action is required.

CKZ42001I hh:mm:ss VOLOPTIONS STARTED - PROGRAM REV=rrr | hh:mm:ss VOLOPTIONS COMPLETED; RETURN CODE=nnn

Explanation:
VOLOPTIONS processing message.

User response:
No action is required.

CKZ42003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:
'ddname' has been dynamically allocated for the indicated data set.

User response:
No action is required.

CKZ42004E DDNAME MISSING: ddname | OPEN FAILED FOR DDNAME: ddname

Explanation:
'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:
Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ42005E ALLOCATION FAILED FOR DSN: datasetname

Explanation:
Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:
If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ42005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ42006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ42007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ42008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ42009E ERROR ACCESSING JOURNAL
FILE; LOC=llll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ42010E DUPLICATE JOURNAL ENTRY;
LOC=llll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42012E JOURNAL CONTROL RECORD IS
WRONG VERSION | JOURNAL
VOLP RECORD IS WRONG
VERSION**

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ42013E RECORD COUNT IS ZERO; LOC=llll
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc**

Explanation:

There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42015E THE COPY PROCESS DID NOT
COMPLETE SUCCESSFULLY**

Explanation:

The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response:

Check that the COPY command has completed successfully before initiating the VOLOPTIONS command.

**CKZ42017E THE NEWTARGETS-DDN IS EMPTY
OR HAS BEEN DUMMIED,
DDNAME: ddn**

Explanation:

No records were read from the ddname specified for CKZMSG NEWTARGETS-DDN. Processing terminates.

User response:

Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ42020I **CURRENT JOURNAL VOLUME PAIRS: | UPDATED JOURNAL VOLUME PAIRS: (** SIMULATION **) list of volume pairs**

Explanation:

The first format indicates the volume pairs at the initiation of the command. The second format indicates the volume pairs after the UPDATE NEWTARGETS have been applied.

User response:

No action is required.

CKZ42021I **UPDATED USERCATALOGS VOLUME: (** SIMULATION **) list of usercatalogs**

Explanation:

Displays the usercatalogs and their volsers after the UPDATE NEWTARGETS have been applied.

User response:

No action is required.

CKZ42031I **VOLUMES FOR NEW TARGETS: source target newtarget**

Explanation:

The listed groups of volume serials were obtained from the NEWTARGETS keyword, or, from the NEWTARGET-DDN file.

User response:

No action is required.

CKZ42040E **UCBLOOK ERROR; RETURN CODE=nn REASON CODE=nn LOC=lllll**

Explanation:

An error occurred during UCBLOOK processing. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ42051E **REQUIRED KEYWORD MISSING: keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ42053E **KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ42054E **KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ42056E **NOTHING SPECIFIED FOR KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ42057E **DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ42058E **INVALID VALUE IN KEYWORD: keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ42060E **RESUME ONLY VALID WITH SOURCECLIP, TARGETOFFLINECLIP, OR TARGETUNCLIP**

Explanation:

The RESUME keyword can only be specified if the SOURCECLIP, TARGETOFFLINECLIP, or TARGETUNCLIP keyword is specified. Processing terminates.

User response:

Correct the keyword specification.

CKZ42063E **INVALID DEVN: devn IN
KEYWORD: keyword**

Explanation:

The volume device number specified is invalid. Processing terminates.

User response:

Correct the device number specification.

CKZ42064E **INVALID VOLSER: volser IN
KEYWORD: keyword**

Explanation:

The volume serial number specified is invalid. Processing terminates.

User response:

Correct the volser specification.

CKZ42066E **VOLSER REFERENCED AS BOTH
SOURCE AND TARGET: volser**

Explanation:

The indicated new target volume serial is already in use as a source volume serial. Processing terminates.

User response:

Correct the volume serial specification.

CKZ42068E **UNMATCHED ENTRIES IN
KEYWORD: keyword**

Explanation:

For NEWTARGETS, there must be a source volume serial, target volume serial, new target volume serial. Unmatched entries were found. Processing terminates.

User response:

Correct the keyword specification.

CKZ42070E **NO MATCH FOUND IN JOURNAL
FOR SOURCE VOLSER: volser**

Explanation:

The indicated volser was specified as a source volume serial in the NEWTARGETS keyword. That source volume serial was not found in the journal records. Processing terminates.

User response:

Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

CKZ42071E **SPECIFIED TARGET DOES NOT
MATCH JOURNAL ENTRY, TARGET:
volser**

Explanation:

The indicated volser was specified as a target volume serial in the NEWTARGETS keyword. That target

volume serial was not found paired to the specified source volume serial. Processing terminates.

User response:

Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

CKZ42072E **SPECIFIED TARGET WILL CAUSE
DUPLICATE TARGET VOLUME
SERIALS, TARGET: volser**

Explanation:

The indicated volser was specified as a target volume serial in the NEWTARGETS keyword. It is either duplicated in the new target volume serials specified, or, will duplicate an existing target volume serial that is not being changed. Processing terminates.

User response:

Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

CKZ42073E **SPECIFIED TARGET WILL CAUSE
DUPLICATE TARGET VOLUME
DEVNS, TARGET DEVN: devn**

Explanation:

The indicated device number was specified as a target volume device number in the NEWTARGETSDEVN keyword. It is either duplicated in the new target volume device numbers specified, or, will duplicate an existing target volume device number that is not being changed. Processing terminates.

User response:

Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

CKZ42101I *hh:mm:ss* VOLOPTIONS
SOURCECLIP |
TARGETOFFLINECLIP STARTED -
PROGRAM REV=*revision* (**
SIMULATION **) | *hh:mm:ss*
VOLOPTIONS SOURCECLIP |
TARGETOFFLINECLIP
COMPLETED; RETURN
CODE=*return_code*

Explanation:

VOLOPTIONS SOURCECLIP or TARGETOFFLINECLIP processing message.

User response:

No action is required.

CKZ42107W **ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ42108E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found.
Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ42109E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file.
Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ42111E JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42136E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser INVALID
PARAMETERS**

Explanation:

The parameters given to IEEVARYD are incorrect.
Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42137E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE:
device RETURN CODE: return_code
REASON CODE: reason_code**

Explanation:

The vary offline for the indicated device failed.
Processing terminates.

User response:

When the problem that caused the vary to fail is corrected, any volumes previously varied offline will need to be varied back online in order to rerun with VOLOPTIONS SOURCECLIP.

**CKZ42138E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE:
device R15: r15_contents**

Explanation:

The vary offline for the indicated device failed.
Processing terminates.

User response:

When the problem that caused the vary to fail is corrected, any volumes previously varied offline will need to be varied back online in order to rerun with VOLOPTIONS SOURCECLIP.

**CKZ42141I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS NOW OFFLINE**

Explanation:

The indicated device is now offline to the current image.

User response:

No action is required.

**CKZ42142I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS PENDING
OFFLINE**

Explanation:

The indicated device is now pending offline to the current image.

User response:

No action is required.

**CKZ42143E SOURCE VOLUME SERIAL: volser
IS OFFLINE AND NO
INFORMATION ABOUT THE
DEVICE IS KNOWN**

Explanation:

The indicated source volume serial is not online and no information about the physical device is known.
Processing terminates.

User response:

Bring the target device with the source volume serial online and run VOLOPTIONS SOURCECLIP with the RESUME keyword.

**CKZ42144E DEVICE NUMBER: devn FOR
TARGET VOLUME: volser DOES
NOT EXIST**

Explanation:

The indicated device number is not defined to the z/OS system. Processing terminates.

User response:

Correct the device number to use a defined device.

**CKZ42145E DEVICE NUMBER: devn IS ONLINE
WITH UNEXPECTED VOLUME
SERIAL: volser**

Explanation:

The indicated device number is online but the volser of the device does not match what is expected. Processing terminates.

User response:

Either correct the device number to use the correct device or correct the device to have the correct contents.

**CKZ42146E DEVICE NUMBER: *device_number*
IS ONLINE; VOLUME SERIAL:
*volser***

Explanation:

The indicated device is online and TARGETOFFLINECLIP was specified. For TARGETOFFLINECLIP, the target device should be offline. Processing terminates.

User response:

Verify that TARGETOFFLINECLIP is really desired. If this is a rerun of a TARGETOFFLINECLIP, the RESUME keyword should be used.

**CKZ42147I VOLUME SERIAL: *volser* DEVICE
NUMBER: *devn* NOT TAKEN
OFFLINE DUE TO SIMULATION**

Explanation:

The indicated volume was not taken offline because this run is a simulation.

User response:

No action is required.

**CKZ42148E SOURCE VOLUME SERIAL= *vvvvvv*
IS OFFLINE AND IS EXPECTED TO
BE ONLINE**

Explanation:

The indicated volume is offline but is expected to be online. Processing terminates.

User response:

If this is a rerun, the RESUME keyword should be used.

**CKZ42160E ERROR DURING UCBxxxx FOR
VOLSER=*volume* - RETURN
CODE=*nnnn* REASON CODE=*nnnn***

Explanation:

An error occurred using UCBLOOK or UCBSCAN. Processing terminates.

User response:

A return code 4 from UCBLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ42179W TARGET VOLUME SERIAL: *volume*
IS CURRENTLY ONLINE**

Explanation:

SOURCECLIP was requested, but, the indicated target volume serial was found online. Processing will continue checking all other targets, but, the run will be terminated.

User response:

Change the target volume serial(s) Db2 Cloning Tool will use with the VOLOPTIONS UPDATE command to avoid duplicate volume serials.

**CKZ42180E NO TARGET VOLSER SHOULD BE
FOUND ONLINE FOR SOURCECLIP
FUNCTION**

Explanation:

SOURCECLIP was requested, but one or more target volumes were found online. Processing terminates.

User response:

See the CKZ42179W message(s) for online target volume serials. If this is a rerun, the RESUME keyword should be used.

**CKZ42201I *hh:mm:ss* VOLOPTIONS
TARGETUNCLIP STARTED -
PROGRAM REV=*revision* (**
SIMULATION **) | *hh:mm:ss*
VOLOPTIONS TARGETUNCLIP
COMPLETED; RETURN
CODE=*return_code***

Explanation:

VOLOPTIONS TARGETUNCLIP processing message.

User response:

No action is required.

**CKZ42203I DDNAME=*ddname* ALLOCATED
FOR DSN=*datasetname***

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ42204E OPEN FAILED FOR DDNAME:
*ddname***

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

**CKZ42205E ALLOCATION FAILED FOR DSN:
*datasetname***

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ42205W DEALLOCATION FAILED FOR
DDNAME: ddname**
Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ42207W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**
Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ42209E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**
Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ42211E JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND**
Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42220E INTERNAL ERROR; INVALID
REQUEST TYPE= type**
Explanation:

An internal error has been encountered. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42235E ICKDSF COMMAND FAILED FOR
TARGET VOLSER: volser**
Explanation:

The invocation of ICKDSF to change a device label failed. The messages from ICKDSF are printed.

User response:

If unable to determine the reason for the failure from the associated ICKDSF messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ42236E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser INVALID
PARAMETERS**
Explanation:

The parameters given to IEEVARYD are incorrect. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42237E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE:
device RETURN CODE: return_code
REASON CODE: reason_code**
Explanation:

The vary offline for the indicated device failed. Processing terminates.

User response:

When the problem that caused the vary to fail is corrected, VOLOPTIONS TARGETUNCLIP can be rerun with the RESUME keyword.

**CKZ42238E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE:
device R15: r15_contents**
Explanation:

The vary offline for the indicated device failed. Processing terminates.

User response:

When the problem that caused the vary to fail is corrected, VOLOPTIONS TARGETUNCLIP can be rerun with the RESUME keyword.

**CKZ42240I DEVICE NUMBER: nnnn
SUCCESSFULLY CHANGED TO
VOLUME SERIAL: volser**
Explanation:

The indicated device has been clipped to the indicated volume serial.

User response:

No action is required.

**CKZ42241I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS NOW OFFLINE |**

**VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS ALREADY
OFFLINE**

Explanation:

The indicated device is now offline or was already offline to the current image.

User response:

No action is required.

**CKZ42242I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS PENDING
OFFLINE**

Explanation:

The indicated device is now pending offline to the current image.

User response:

No action is required.

**CKZ42243E TARGET VOLUME SERIAL: volser IS
OFFLINE AND NO INFORMATION
ABOUT THE DEVICE IS KNOWN**

Explanation:

The indicated target volume serial is not online and no information about the physical device is known. Processing terminates.

User response:

Bring the target device with the target volume serial online and rerun VOLOPTIONS TARGETUNCLIP with the RESUME keyword.

**CKZ42244E TARGET VOLUME SERIAL= vvvvvv
IS OFFLINE AND IS EXPECTED TO
BE ONLINE**

Explanation:

The indicated volume is offline but is expected to be online. Processing terminates.

User response:

If this is a rerun, the RESUME keyword should be used.

**CKZ42245I DEVICE NUMBER: devn ALREADY
CHANGED TO SOURCE VOLUME
SERIAL: volser**

Explanation:

The indicated device number has already been clipped to the desired volume serial.

User response:

No action is required.

**CKZ42246I DEVICE NUMBER: nnnn ICKDSF
FAILED; DEVICE MAY HAVE
ALREADY BEEN UNCLIPPED**

Explanation:

The unclip of the indicated device by ICKDSF has failed. This failure is probably caused by the device having already been unclipped.

User response:

See the next Db2 Cloning Tool message in the listing to determine the appropriate action.

**CKZ42247I VOLUME SERIAL: volser DEVICE
NUMBER: devn NOT TAKEN
OFFLINE DUE TO SIMULATION**

Explanation:

The indicated volume was not taken offline because this run is a simulation.

User response:

No action is required.

**CKZ42248I VOLUME SERIAL: vvvvvv DEVICE
NUMBER: nnnn NOT CLIPPED TO
VOLUME SERIAL: vvvvvv DUE TO
SIMULATION**

Explanation:

The clip of the indicated device was not done because this run is a simulation.

User response:

No action is required.

**CKZ42260E ERROR DURING UCBxxxx FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using UCBLLOOK or UCBSKAN. Processing terminates.

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ43001I hh:mm:ss UCATOPTIONS
STARTED - PROGRAM REV=rrr |
hh:mm:ss UCATOPTIONS
COMPLETED; RETURN CODE=nnn**

Explanation:

UCATOPTIONS processing message.

User response:

No action is required.

**CKZ43003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ43004E DDNAME MISSING: ddname |
OPEN FAILED FOR DDNAME:
ddname**

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

**CKZ43005E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ43005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ43006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ43007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ43008E UNABLE TO LOAD PROGRAM:
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ43009E ERROR ACCESSING JOURNAL
FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ43010E DUPLICATE JOURNAL ENTRY;
LOC=lllll**

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ43011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL UCAT PAIR
RECORD(S) NOT FOUND**

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ43012E JOURNAL CONTROL RECORD IS
WRONG VERSION | JOURNAL
UCAT PAIR RECORD IS WRONG
VERSION**

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ43013E RECORD COUNT IS ZERO; LOC=lllll
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc**

Explanation:

There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ43015E THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY

Explanation:

The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response:

Check that the COPY command has completed successfully before initiating the VOLOPTIONS command.

CKZ43017E THE NEWTARGETS-DDN IS EMPTY OR HAS BEEN DUMMIED, DDNAME: ddn

Explanation:

No records were read from the ddname specified for NEWTARGETS-DDN. Processing terminates.

User response:

Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ43018E THE NEWTARGETS-DDN INPUT HAS EXCEEDED THE CURRENT CAPACITY

Explanation:

The number of entries read from the ddname exceeded the current capacity. Processing terminates.

User response:

The UCATOPTIONS UPDATE may be run multiple times with the input split into smaller parts. In addition, please report this message to IBM Software Support.

CKZ43020I CURRENT JOURNAL UCAT PAIRS: | UPDATED JOURNAL UCAT PAIRS:

Explanation:

The first format indicates the user catalog pairs at the initiation of the command. The second format indicates the user catalog pairs after the UPDATE NEWTARGETS have been applied.

User response:

No action is required.

CKZ43031I ENTRIES FROM NEWTARGETS: source newtarget | ENTRIES FROM

NEWCATWORKS: current-dsn new-dsn

Explanation:

The listed pairs of catalog names were obtained from the NEWTARGETS keyword, or, from the NEWTARGETS-DDN file. Or, the listed pairs of data set names were obtained from the NEWCATWORKS keyword, or, from the NEWCATWORKS-DDN file.

User response:

No action is required.

CKZ43040E SOURCE CATALOG BACKUP HAS NOT BEEN DONE

Explanation:

The source catalogs have not been backed up. Processing terminates.

User response:

Run UCATOPTIONS BACKUP to backup the source catalogs.

CKZ43041E SOURCE CATALOG BACKUP HAS ALREADY BEEN DONE

Explanation:

The source catalogs have already been backed up. Processing terminates.

User response:

UCATOPTIONS BACKUP does not need to be run.

CKZ43048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how UCATOPTIONS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ43051E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZ43052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token

Explanation:

An error occurred validating the CKZINI member options. Processing terminates.

User response:

Correct the CKZINI member.

CKZ43053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

CKZ43054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

CKZ43056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

CKZ43057E DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

CKZ43058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

CKZ43064E INVALID UCAT: bcs dsname IN KEYWORD: keyword | INVALID CATWORK: dsname IN KEYWORD: keyword

Explanation:

The indicated bcs dsname or dsname is invalid. Processing terminates.

User response:

Correct the invalid dsname.

CKZ43065E FORCE CAN ONLY BE SPECIFIED WITH BACKUP

Explanation:

The FORCE keyword was specified, but the BACKUP keyword was not specified. The FORCE keyword can only be specified with the BACKUP keyword. Processing terminates.

User response:

Correct the keyword specification.

CKZ43068E UNMATCHED ENTRIES IN KEYWORD: keyword

Explanation:

For NEWTARGETS, there must be a source user catalog and a new target user catalog. Unmatched entries were found. Processing terminates.

User response:

Correct the keyword specification.

CKZ43073E NO MATCH FOUND IN JOURNAL FOR SOURCE UCAT: codename | NO MATCH FOUND IN JOURNAL FOR CATWORK DSN: dsname

Explanation:

The indicated user catalog was specified as a source in the NEWTARGETS keyword. That source usercatalog was not found in the journal records. Or, the indicated dsname was specified as a current value in the NEWCATWORKS keyword. That catwork dsname was not found in the journal records. Processing terminates.

User response:

Correct the keyword specification. The UCATOPTIONS LIST command may be used to display the ucat pairs in the journal.

CKZ43074W NEWTARGETS IGNORED WITH "LIST" OPTION | NEWCATWORKS IGNORED WITH "LIST" OPTION

Explanation:

NEWTARGETS or NEWCATWORKS was specified with UCATOPTIONS LIST. The NEWTARGETS or NEWCATWORKS keyword is ignored. Processing continues.

User response:

None, unless UPDATE was intended.

CKZ44001I hh:mm:ss ONLINECLIP STARTED - PROGRAM REV=rrr | hh:mm:ss ONLINECLIP COMPLETED; RETURN CODE=nnn

Explanation:

ONLINECLIP command processing message.

User response:

No action is required.

CKZ44003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ44004E DDNAME MISSING: ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ44005E ALLOCATION FAILED FOR DSN: datasetname

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ44005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ44007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ44009E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation:

A VSAM error occurred accessing the journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ44011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLUME PAIR RECORD(S) NOT FOUND

Explanation:

An expected journal record was not found or did not match the expected format. If the format is the problem, the record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ44013E RECORD COUNT IS ZERO; LOC=lllll | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation:

There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ44015E THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY

Explanation:

The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response:

Check that the COPY command has completed successfully before initiating the ONLINECLIP command.

CKZ44033E EXCP FAILED FOR DEVICE: nnnn TARGET VOLSER: volser SYNAD TEXT: text

Explanation:

An error occurred accessing the volume label for a target device. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZ44034W VOLUME SERIAL FOR DEVICE:
nnnn IS volser; VOLUME SERIAL
EXPECTED IS sourcevolser**

Explanation:

The internal volume label for the device is 'volser'. The program expected the internal label to match the associated source volume serial. Processing continues.

User response:

If the internal volume serial is already the target volume serial, there should be no problem. However, if the internal serial number is not related to the current Db2 Cloning Tool process, check that the volume pairs given to the Db2 Cloning Tool COPY command were correct.

**CKZ44035I VOLUME SERIAL FOR DEVICE:
nnnn CHANGED TO: targetvolser**

Explanation:

The internal volume label for the device has been changed to the target volume serial.

User response:

No action is required.

**CKZ44048I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation:

Informational message indicating how ONLINECLIP will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

**CKZ44050E ERROR IN PARAMETERS FOR
keyword**

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ44051E REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ44053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ44054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ44056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ44058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ44060E ERROR DURING function FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using IOSCAPU or UCBLLOOK. Processing terminates.

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ44079E TARGET VOLSER: targetvolser
WAS NOT FOUND ONLINE**

Explanation:

The indicated target volume serial was not found online. Processing terminates.

User response:

The target volumes are expected to be online for ONLINECLIP. Correct the problem with the target volume(s).

CKZ44086I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

**CKZ45001I hh:mm:ss VARYOFF STARTED -
PROGRAM REV=rrr | hh:mm:ss
VARYOFF COMPLETED; RETURN
CODE=nnn**

Explanation:

VARYOFF command processing message.

User response:

No action is required.

**CKZ45003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

**CKZ45004E DDNAME MISSING=ddname |
OPEN FAILED FOR
DDNAME=ddname**

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

**CKZ45005E ALLOCATION FAILED FOR
DSN=datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ45005W DEALLOCATION FAILED FOR
DDNAME=ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ45006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ45007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ45008E UNABLE TO LOAD PROGRAM=
program name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ45009E ERROR ACCESSING JOURNAL
FILE; LOC=lllll | ERROR
ACCESSING VARY FILE; LOC=lllll**

Explanation:

A VSAM error occurred accessing the journal or vary file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ45010E DUPLICATE VARY ENTRY; LOC=lllll

Explanation:

A duplicate record was detected. Processing terminates.

User response:

Verify that an empty vary file was used as input to the VARYOFF command. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ45011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL VOLP RECORD NOT FOUND

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45012E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLP RECORD IS WRONG VERSION

Explanation:

The journal record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ45013E COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation:

There was a problem with the journal records needed to initiate the command. The number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45017E THE DDNAME IS EMPTY OR HAS BEEN DUMMIED, DDNAME: ddn

Explanation:

No records were read from the ddname specified for a keyword. Processing terminates.

User response:

Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ45018E THE DDNAME INPUT HAS EXCEEDED THE CURRENT CAPACITY, DDNAME: ddn

Explanation:

The number of entries read from the ddname exceeded the current capacity. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45019E DDNAME: ddn DOES NOT HAVE A LRECL OF 80

Explanation:

The data set allocated to the ddname does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response:

Change the data set allocated to the ddname to have a LRECL of 80.

CKZ45020I COPY COMMAND: copy command

Explanation:

Lists the cop command.

User response:

No action is required.

CKZ45028E ERROR CREATING COMMAND STRING; LOC=location

Explanation:

This is an internal error. A problem occurred in the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45031E DEVICE NOT DEFINED; VOLSER: volser DEVICE NUMBER: dddd

Explanation:

The device number that was specified is not defined to the z/OS system. Processing terminates.

User response:

Change the specified device number to be a defined device defined to the z/OS system.

CKZ45032E VOLUME ALREADY OFFLINE; VOLSER: volser DEVICE NUMBER: dddd

Explanation:

The device number that was specified is already offline to the z/OS system. Processing terminates if VOL-ALREADY-OFFLINE(QUIT) is specified.

User response:

Bring the volume online or use VOL-ALREADY-OFFLINE(CONTINUE) if the volume should already be offline.

CKZ45033E WRONG VOLSER FOUND; FOUND VOLSER: volser EXPECTED

**VOLSER: volser DEVICE NUMBER:
dddd**

Explanation:

The wrong volser was found on device dddd.
Processing terminates.

User response:

Correct the volume and device number specifications
to match the current volser on the device.

**CKZ45034E VOLUME NOT ONLINE AND DEVICE
NUMBER IS NOT KNOWN;
VOLSER: volser**

Explanation:

The volume is not online and a device number has not
been specified for the volume. Processing terminates.

User response:

Either bring the volume online or specify the device
number of the volume if possible.

**CKZ45035E DEVICE NOT DASD; VOLSER:
volser DEVICE NUMBER: dddd**

Explanation:

The specified device is not defined to z/OS as a DASD
device. Processing terminates.

User response:

Change the device number to be that of a defined
DASD device.

**CKZ45040E CKZ00900 UNEXPECTED
RESULTS; error text**

Explanation:

An unexpected condition occurred calling program
CKZ00900. 'error text' has a description of the
problem. Processing terminates.

User response:

Contact IBM Software Support. Have available the
listing that contains this message.

**CKZ45041E INVALID VALUE FOUND FOR item
IN keyword RECORD: value**

Explanation:

An invalid value has been found for an item in a record
in the data set allocated to the ddname for the
keyword. The record is printed. Processing terminates.

User response:

Correct the value for the item in the record to have a
valid value.

**CKZ45048I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation:

Informational message indicating how VARYOFF will
handle the options. The displayed options are derived
from the CKZINI and any overriding specifications in
the command input.

User response:

No action is required.

**CKZ45050E ERROR IN PARAMETERS FOR
keyword**

Explanation:

The parameters for the indicated keyword were
incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive
keywords may have been used.

**CKZ45051E REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted.
Processing terminates.

User response:

Specify the required keyword.

**CKZ45053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the
maximum length allowed for the operand. nnn is the
maximum allowed length for the keyword. Processing
terminates.

User response:

Correct the length of the keyword's operand.

**CKZ45054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only
one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ45055E STORAGE GROUPS NOT
SUPPORTED WITH DATA-MOVER
PROGRAM NONE**

Explanation:

The DATA-MOVER program was specified as 'NONE'.
Storage group names/masks were specified for the
source and/or target volume serials. Processing
terminates.

User response:

Correct the DATA-MOVER program specified, or, use
keywords FROM-VOLSER/TO-VOLSER for the volume
serials.

**CKZ45056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ45057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation:

The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response:

Remove the duplicate entry.

**CKZ45058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ45060E function ERROR; RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using the UCBLook or UCBScan macro. Processing terminates.

User response:

A return code 4 from UCBLook may indicate the volsr is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ45061E CKZ01SMF ERROR; RETURN
CODE=nnnn LOC: llllll entry**

Explanation:

An error occurred using CKZ01SMF to obtain SSI information for the 'entry'. llllll is the internal location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

**CKZ45062E NO STORAGE GROUPS RETURNED
BY SSI**

Explanation:

The VARYOFF command was requested to determine volume serial numbers by storage group names. SSI did not return any storage groups to CKZ01SMF. Processing terminates.

User response:

Check that storage groups are defined on the system. If unable to determine the cause of this message, contact IBM Software Support. Have available the listing containing this message.

**CKZ45063W NO STORAGE GROUP MATCH
FOUND; ENTRY: entry | NO
VOLUME MATCH FOUND; ENTRY:
entry**

Explanation:

The indicated 'entry' for the keyword was not matched. For STORAGE GROUP, no matches were found for the mask. For VOLUME, the indicated volsr, or, the volsr derived from a storage group, was not found. Processing continues.

User response:

No action is required.

**CKZ45063E EXPLICIT STORAGE GROUP NOT
FOUND; ENTRY: entry**

Explanation:

The indicated storage group name is not defined as a storage group. Processing terminates.

User response:

Correct to specify defined storage group name.

**CKZ45067W NO VOLUME SERIALS FOR
STORAGE GROUP: storage group**

Explanation:

SSI did not return any volsers for the storage group to CKZ01SMF. Processing continues.

User response:

No action is required.

**CKZ45070E THE COPY COMMAND HAS
UNBALANCED PARENTHESES**

Explanation:

The copy command has unbalanced parentheses. Processing terminates.

User response:

Correct the copy command to have balanced parentheses.

**CKZ45071E THE COPY COMMAND HAS A
KEYWORD THAT IS TOO LONG:
keyword**

Explanation:

The copy command has a keyword specified that is longer than a keyword is expected to be. Processing terminates.

User response:

Correct the copy command keyword to be valid.

**CKZ45072E THE COPY COMMAND IS NOT A
COPY COMMAND**

Explanation:

The copy command does not appear to be a copy command. Processing terminates.

User response:

Correct the copy command.

CKZ45073I PROCESSING COPY COMMAND

Explanation:

The copy command is being read and parsed.

User response:

No action is required.

**CKZ45074E COPY COMMAND HAS MUTUALLY
EXCLUSIVE KEYWORDS:
keyword1 keyword2**

Explanation:

The copy command has keywords specified that are mutually exclusive. Processing terminates.

User response:

Correct the copy command to not have mutually exclusive keywords specified.

**CKZ45075E UNMATCHED ENTRIES IN COPY
COMMAND KEYWORD: keyword**

Explanation:

For VOLPAIRS, there must be a source volume serial, target volume serial. An uneven number of entries was specified. For VOLPAIRSDEVN, there must be a source volume serial, target volume serial, target device number. Unmatched entries were found. Processing terminates.

User response:

Correct the keyword specification.

CKZ45076E NO VOLUMES SERIALS SELECTED

Explanation:

No volumes were selected for processing. Processing terminates.

User response:

Correct the copy command so volumes will be selected for processing.

**CKZ45077I VOLUMES SELECTED FOR VARY
OFFLINE PROCESSING: volser
dddd**

Explanation:

Lists the volser and device number of the volumes selected for processing.

User response:

No action is required.

**CKZ45078E VOLUME PAIRS ONLY ACCEPTED
WITH DATA-MOVER PROGRAM
NONE**

Explanation:

Keyword VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN was used, but, the

DATA-MOVER program is not 'NONE'. Processing terminates.

User response:

Correct the keyword specifications.

CKZ45086I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

**CKZ45101I hh:mm:ss VARY VOLUMES
STARTED - PROGRAM REV=rrr |
hh:mm:ss VARY VOLUMES
COMPLETED; RETURN CODE=nnn**

Explanation:

VARYON command processing message.

User response:

No action is required.

**CKZ45106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation:

A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ45107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ45108E UNABLE TO LOAD PROGRAM:
program_name**

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ45120I CONSOLE name ACQUIRED FOR
SYSPLEX VARY COMMANDS**

Explanation:

CKZ45138E **IEEVARYD VARY OFFLINE FAILED FOR VOLSER: volser DEVICE: dddd R15: nnnnnnnn | IEEVARYD VARY ONLINE FAILED FOR VOLSER: volser DEVICE: dddd R15: nnnnnnnn**

Explanation:

The vary for the indicated device failed. Processing terminates.

User response:

Correct the problem causing the vary to fail. If unable to correct the problem, Contact IBM Software Support. Have available the listing that contains this message.

CKZ45140I **NO VOLUMES NEEDED LOCAL VARY OFFLINE | NO VOLUMES NEEDED LOCAL VARY ONLINE**

Explanation:

No volumes needed to be varied on the local system.

User response:

No action is required.

CKZ45141I **VOLUME SERIAL: volser DEVICE NUMBER: dddd IS NOW OFFLINE | VOLUME SERIAL: volser DEVICE NUMBER: dddd IS NOW ONLINE**

Explanation:

The indicated device is now offline or online to the current system.

User response:

No action is required.

CKZ45144E **DEVICE NUMBER: dddd FOR TARGET VOLUME: volser DOES NOT EXIST**

Explanation:

The indicated device number is not defined to the z/OS system. Processing terminates.

User response:

Correct the device number to use a defined device.

CKZ45145E **DEVICE NUMBER: dddd IS ONLINE WITH UNEXPECTED VOLUME SERIAL: volser**

Explanation:

The indicated device number is online but the volser of the device does match what is expected. Processing terminates.

User response:

Either correct the device number to use the correct device or correct the device to have the correct contents.

CKZ45146W **DEVICE NUMBER: dddd FOR VOLUME: volser IS STILL OFFLINE**

Explanation:

The vary online got a good return code but the device is still offline. Processing continues.

User response:

Determine why the volume went offline after it was varied online. If unable to determine the cause, contact IBM Software Support. Have available the listing containing this message and the system log from the time this happened.

CKZ45147I **VOLUME SERIAL: volser DEVICE NUMBER: dddd WILL BE VARIED OFFLINE | VOLUME SERIAL: volser DEVICE NUMBER: dddd WILL BE VARIED ONLINE**

Explanation:

Display the vary that will be done.

User response:

No action is required.

CKZ45160E **ERROR DURING UCBLLOOK FOR DEVICE NUMBER dddd VOLUME SERIAL: volser RETURN CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using the UCBLLOOK macro. Processing

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ45201I **hh:mm:ss MVS ROUTE COMMAND PROCESSOR STARTED REV=revision; COMMAND: command | hh:mm:ss MVS ROUTE COMMAND PROCESSOR COMPLETED. COMMAND: command RETURN CODE=return_code**

Explanation:

This message provides information on the state of the MVS ROUTE command processing.

User response:

No action is required.

CKZ45204E **WRONG COMMAND. STATE: state COMMAND: command WORK AREA ADDRESS: address**

Explanation:

A command in the wrong format has been sent to the MVS ROUTE command processor.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45205E **EMPTY COMMAND STRING.
ROUTE STRING CANNOT BE
PREPARED**

Explanation:

An empty command string was sent to the MVS ROUTE command processor.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45207W **ERROR CALLING CKZ01HEX;
FUNCTION: *function*
R15=*return_code***

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45209E **UNABLE TO LOAD MODULE:
IEANTRT**

Explanation:

The program module IEANTRT was not found. Processing terminates.

User response:

Ensure that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ45211E **ERROR GETTING SYSTEM GROUP
DATA. RETURN CODE: *return_code*
REASON: *reason***

Explanation:

An error getting the system group data was encountered. Processing terminates.

User response:

If the reason is A SYSTEM ERROR OCCURRED 10 TIMES, the command should be repeated. If the reason is THE SYSTEM CANNOT FIND THE REQUESTED SYSTEM GROUP BY NAME/TOKEN, verify that the required NAME/TOKEN was defined for this system group. If the reason is A NAME/TOKEN PAIR CREATED BY AN UNAUTHORIZED CALLER, verify that the required authorizations exist for working with NAME/TOKEN. If unable to resolve the problem, contact IBM Software Support.

CKZ45212E **INCORRECT TOKEN FORMAT FOR
SYSTEM GROUP NAME:
group_name REASON: *reason_code*
TOKEN: *token_value***

Explanation:

A required system group name was found, but the format of the TOKEN is incorrect. Processing terminates.

User response:

Verify that the format of NAME/TOKEN is defined in accordance with SYS1.SAMPLIB(IEEGSYS). If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ45213E **FIRST MEMBER OF SYSTEMS LIST
HAS INCORRECT FORMAT**

Explanation:

The first member in the systems list is not 'GSYS'. Processing terminates.

User response:

Verify that the format of NAME/TOKEN is defined in accordance with SYS1.SAMPLIB(IEEGSYS). If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ45215E **INVALID TIME INTERVAL FOR
ROUTE**

Explanation:

An invalid time interval format for the MVS ROUTE command was encountered. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45216E **NO PREPARED ROUTE COMMAND
FOR EXECUTING**

Explanation:

No route command was prepared for the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ45217I **SYSTEM GROUP: *group_name*
CONTAINS ONLY LOCAL SYSTEM:
system_name. THERE IS NO
POSSIBILITY TO EXECUTE ROUTE
ON OTHER SYSTEMS. COMMAND:
*command***

Explanation:

Only the local system is defined in the system group. It is not possible to run the ROUTE command on other systems in this group. Processing continues.

User response:

No action is required.

CKZ45218I **THE LOCAL SYSTEM: *system_name*
IS DEFINED IN THE SYSTEM
GROUP: *group_name* | THE LOCAL
SYSTEM: *system_name* IS NOT**

DEFINED IN THE SYSTEM GROUP:
group_name

Explanation:

This message indicates whether the local system *system_name* is defined in the system group *group_name*.

User response:

No action is required.

CKZ45220I ROUTE COMMAND FOR EXECUTION: *command*

Explanation:

The command *command* has been prepared and will be run by the MVS ROUTE command processor.

User response:

No action is required.

CKZ46001I hh:mm:ss VARYON STARTED - PROGRAM REV=rrr | hh:mm:ss VARYON COMPLETED; RETURN CODE=nnn

Explanation:

VARYON command processing message.

User response:

No action is required.

CKZ46003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ46004E DDNAME MISSING=ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ46005E ALLOCATION FAILED FOR DSN: datasetname

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ46005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ46007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ46008E UNABLE TO LOAD PROGRAM: program name

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ46009E ERROR ACCESSING JOURNAL FILE; LOC=lllll | ERROR ACCESSING VARY FILE; LOC=lllll

Explanation:

A VSAM error occurred accessing the journal or vary file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ46011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL VOLP RECORD NOT FOUND | VARY CONTROL RECORD NOT FOUND | VARY VOL RECORD NOT FOUND

Explanation:

An expected record was not found in the Db2 Cloning Tool journal file or the Db2 Cloning Tool vary file. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ46012E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLP RECORD IS WRONG VERSION | VARY CONTROL RECORD IS WRONG VERSION | VARY VOL RECORD IS WRONG VERSION

Explanation:

The journal or vary record does not match the expected format. The record is printed. Processing terminates.

User response:

Verify that different releases of Db2 Cloning Tool have not been run using the same journal or vary data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ46013E COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation:

There was a problem with the journal or vary records needed to initiate the command. The number of records read from the journal or vary, rrrr, is not the same as the number indicated in the journal or vary control record, cccc. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ46028E ERROR CREATING COMMAND STRING; LOC=location

Explanation:

This is an internal error. A problem occurred in the MVS ROUTE command processor. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ46031E DEVICE NOT DEFINED; VOLSER: volser DEVICE NUMBER: dddd

Explanation:

The device number that was specified is not defined to the z/OS system. Processing terminates.

User response:

Change the specified device number to be a defined device defined to the z/OS system.

CKZ46032E VOLUME ALREADY ONLINE; VOLSER: volser DEVICE NUMBER: dddd

Explanation:

The device number that was specified is already online to the z/OS system. Processing terminates if VOL-ALREADY-ONLINE(QUIT) is specified.

User response:

Bring the volume offline or use VOL-ALREADY-ONLINE(CONTINUE) if the volume should already be online.

CKZ46033E WRONG VOLSER FOUND; FOUND VOLSER: volser EXPECTED VOLSER: volser DEVICE NUMBER: dddd

Explanation:

The wrong volser was found on device devn. Processing terminates.

User response:

Correct the volume and device number specifications to match the current volser on the device.

CKZ46034E DEVICE NUMBER IS NOT KNOWN; VOLSER: volser

Explanation:

The volume is not online and a device number has not been specified for the volume. The device number is needed to vary the volume online. Processing terminates.

User response:

Either bring the volume online or specify the device number of the volume if possible.

CKZ46048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation:

Informational message indicating how VARYON will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response:

No action is required.

CKZ46050E ERROR IN PARAMETERS FOR keyword

Explanation:

The parameters for the indicated keyword were incorrect. Processing terminates.

User response:

Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ46051E REQUIRED KEYWORD MISSING: keyword

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ46053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ46054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ46056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ46058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ46060E function ERROR; RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation:

An error occurred using the UCBLLOOK macro. Processing terminates.

User response:

A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ46077I VOLUMES SELECTED FOR VARY
ONLINE PROCESSING: volser
dddd**

Explanation:

Lists the volser and device number of the volumes selected for processing.

User response:

No action is required.

CKZ46086I VALIDATING KEYWORD: keyword

Explanation:

Parsing is checking the indicated keyword indicated in the command.

User response:

No action is required.

**CKZ48001I hh:mm:ss JRNLUPGRADE
STARTED - PROGRAM REV=rrr |
hh:mm:ss JRNLUPGRADE
COMPLETED; RETURN CODE=nnn**

Explanation:

JRNLUPGRADE command processing message.

User response:

No action is required.

**CKZ48003I DDNAME=ddname ALLOCATED
FOR DSN=datasetname**

Explanation:

'ddname' has been dynamically allocated for the indicated data set.

User response:

No action is required.

CKZ48004E DDNAME MISSING=ddname

Explanation:

'ddname' was specified for Db2 Cloning Tool to use. Processing terminates.

User response:

Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

**CKZ48005E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation:

Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ48005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation:

Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response:

If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ48007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

**CKZ48009E ERROR ACCESSING JOURNAL
FILE; DDN=ddname LOC=lllll**

Explanation:

A VSAM error occurred accessing a journal file. Processing terminates.

User response:

See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ48010E DUPLICATE JOURNAL ENTRY;
DDN=ddname LOC=lllll**

Explanation:

A duplicate record was detected in the NEW journal file. Processing terminates.

User response:

Verify the referenced NEW journal file was empty when the JRNLUPEGRADE command started processing. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ48020W UNKNOWN JOURNAL RECORD
TYPE: type**

Explanation:

The OLD journal contains an unknown record type. The record is copied to the NEW journal. Not all journal records need to be upgraded for the scenarios documented where JRNLUPEGRADE can be used.

User response:

Verify the current version of the JRNLUPEGRADE command is being used. Please report this message to IBM Software Support.

**CKZ48021E UNKNOWN JOURNAL RECORD
VERSION**

Explanation:

The OLD journal contains an unknown version of a record. The record is printed. Processing terminates.

User response:

Verify the current version of the JRNLUPEGRADE command is being used. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ48031I nnn TOTAL RECORDS READ

Explanation:

Shows the total number of records read from the OLD journal.

User response:

No action is required.

CKZ48032I nnn TOTAL RECORDS WRITTEN

Explanation:

Shows the total number of records written to the NEW journal.

User response:

No action is required.

CKZ48033I nnn type RECORDS UPGRADED

Explanation:

Shows the total number of records of the displayed type that have been upgraded to the current level.

User response:

No action is required.

**CKZ48034I nnn type RECORDS ALREADY AT
CURRENT LEVEL**

Explanation:

Shows the total number of records of the displayed type that did not need to be upgraded because they were already at the current level.

User response:

No action is required.

CKZ48035I nnn UNKNOWN RECORDS

Explanation:

Shows the total number of records of unknown type that were copied without change to the NEW journal.

User response:

No action is required.

**CKZ48051E REQUIRED KEYWORD MISSING:
keyword**

Explanation:

A keyword required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

**CKZ48053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation:

The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the

maximum allowed length for the keyword. Processing terminates.

User response:

Correct the length of the keyword's operand.

**CKZ48054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation:

Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response:

Correct the keyword to use one operand.

**CKZ48056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation:

A keyword was entered without an appropriate operand. Processing terminates.

User response:

Specify an appropriate operand for the keyword.

**CKZ48058E INVALID VALUE IN KEYWORD:
keyword VALUE: value**

Explanation:

The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response:

Correct the value specified in the keyword.

**CKZ50001E PRODUCT INIT FAILURE, RC=rrr,
RS=sss**

Explanation:

Initialization has failed. rrr = return code sss = reason code

User response:

Find one or more detailed warning or failure messages. Use the more detailed message(s) to determine the cause.

CKZ50002I Initialization complete

Explanation:

Initialization has completed successfully.

User response:

No action is required.

**CKZ50003I TCP SERVER JOB, NUMBER OF
SUBTASKS RESET TO 1**

Explanation:

MAX_SUBTASKS in PARMLIB set to a value greater than 1 when starting a TCPIP server job. The value has been changed to a 1.

User response:

No action is required.

CKZ50004E DISCOVERY PHASE HAS FAILED

Explanation:

One or more errors has occurred during the source job discovery phase. The job terminates.

User response:

Find one or more detailed warning or failure messages. Use the more detailed message(s) to determine the cause.

**CKZ50005W DISCOVERY PHASE ENDED WITH
WARNING(S)**

Explanation:

One or more warning messages were issued during the source job discovery phase.

User response:

Find one or more detailed warning messages. Use the more detailed message(s) to determine the cause and correct if needed.

**CKZ50006E SUBTASK nn, IS HUNG, UNABLE
TO EXIT**

Explanation:

The hang may or may not be an internal problem. nn = subtask number

User response:

Contact IBM Software Support unless the hang is the result of an MVS or DB2 problem.

**CKZ50007E UNEXPECTED ATTACH ERROR,
RC=rrr, RS=sss rrr = return code
sss = reason code**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ50008E UNABLE TO ALLOCATE A TDE

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ50009E UNABLE TO ATTACH SUBTASK

Explanation:

This is probably an internal error.

User response:

Contact IBM Software Support.

CKZ50010E QUEUE SHOW ERROR, RS=sss

Explanation:

This is an internal error. sss = reason code

User response:

Contact IBM Software Support.

**CKZ50011I PROCESSING COMPLETE WITH
NO ERRORS**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ50012E COMPLETED, WITH ERRORS,
RC=rrr, RS=sss**

Explanation:

CKZ completed but with errors. rrr = return code sss = reason code

User response:

Find one or more detailed failure messages. Use the more detailed message(s) to determine the cause.

**CKZ50013W Completed, with warnings,
RC=return_code, RS=reason_code**

Explanation:

Db2 Cloning Tool completed but with warnings.

User response:

Find one or more detailed warning messages. Use the more detailed message(s) to determine the cause.

**CKZ50014E Invalid panel request,
request_field**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ50015I Panel terminate request

Explanation:

This message is informational.

User response:

No action is required.

CKZ50016I mmm ddd ttt VERS=vvv nnn...nnn

Explanation:

Module level. mmm = LMOD name ddd = assembly date ttt = assembly time vvv = version nnn = fields unused

User response:

No action is required.

**CKZ50017E Highest return code exceeds MAX-
RC value**

Explanation:

This message indicates that a Db2 Cloning Tool job ended with an error because it encountered a return code that is greater than maximum allowed return code, which is controlled by MAX-RC keyword. When MAX-RC is set to 4, only errors result in job

termination. When MAX-RC is set to 0, both error and warning messages can result in job termination.

User response:

Review other Db2 Cloning Tool messages to determine the reason for the failure.

**CKZ50018E When SUBTASK-DATASET-
EXTENSIONS is Y, feature_name
cannot be enabled.**

Explanation:

The functionality listed in the message is not compatible with the SUBTASK-DATASET-EXTENSIONS(Y) parameter and cannot be enabled.

User response:

Disable the feature listed in the message and resubmit the job.

**CKZ50019I STOP command accepted, job
started_task_name, jobID
job_id_for_started_task.**

Explanation:

The started task specified with the job name parm is processing the requested STOP command.

User response:

No action is required.

CKZ50090I INITIALIZATION COMPLETE: sss

Explanation:

This is an informational WTO. sss = EXEC module name, version and release, assembly date and time

User response:

No action is required.

CKZ50091E INITIALIZATION FAILED

Explanation:

This is an error WTO.

User response:

Check detailed error message(s) for the reason. Check the ACTION field of the detailed message(s).

CKZ50092E ERROR TAKING SVC DUMP: rrr

Explanation:

This is an error WTO. rrr = R15 return from SDUMP

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ50093E ERROR TAKING SVC DUMP: rrr

Explanation:

This is an error WTO. This SVC was attempted from the ESTAE routine rrr = R15 return from SDUMP

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ50094E ERROR TAKING SVC DUMP: rrr

Explanation:

This is an error WTO. This SVC was attempted from the ESTAE RETRY routine. rrr = R15 return from SDUMP

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ50095I NO ESTAE RETRY ALLOWED PER SDWA

Explanation:

This is an informational WTO.

User response:

No action is required.

CKZ50098E *message text*

Explanation

This is an error WTO from initialization. It is followed by CKZ50099E.

message text = error text:

- allocate BVT failed
- allocate save areas failed
- parameter parsing error
- parameter value error, see CKZPRINT
- length and/or count parm exceeds 32k-1
- error return from INIIALIZATION
- ESTAE non zero RC

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ50099E INITIALIZATION FAILED

Explanation:

This is an error WTO. It follows WTO CKZ50098E.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ50601I DSS LEVEL IS hhh, ttt SUPPORT FASTREPLICATE

Explanation:

This message is informational. hhh = DSS level ttt = Does or Does Not

User response:

If the text is Does Not, then DSS will initiate normal copies without using FASTREPLICATE.

CKZ50602E ANTMMAIN IS NOT ACTIVE

Explanation:

This is a system environment error. ANTMMAIN is the System Data Mover.

User response:

Contact your system programmer.

CKZ50603E ANTRQST ERROR: ttt RC=rrr, RS=sss, VER=lll

Explanation

This is a system environment error. ANTMMAIN is the System Data Mover.

- ttt = VERSION of DATA-MOVER
- rrr = return code
- sss = reason code
- lll = the level of ANTMMAIN

User response:

Contact your system programmer.

CKZ50604W ANTRQST LEVEL lll IS NOT SUPPORTED

Explanation:

This is a system environment error. ANTMMAIN is the System Data Mover. The level returned was unexpected. lll = the level of ANTMMAIN

User response:

Contact your system programmer.

CKZ50605I ANTRQST LEVEL IS lll

Explanation:

This message is informational. lll = the level of ANTMMAIN

User response:

No action is required.

**CKZ50606I User private region statistics:
size=size_of_user_region_below_16
_MB KB,
number_of_allocated_kilobytes
_in_user_region_below_16MB KB
allocated,
number_of_free_kilobytes
_in_user_region_below_16 MB KB
free. User extended private region
statistics:
size=size_of_user_region_above_16
_MB KB,
number_of_allocated_kilobytes
_in_user_region_above_16MB KB
allocated,
number_of_free_kilobytes
_in_user_region_above_16 MB KB
free**

Explanation:

This informational message shows the total amount of storage in the user region, as well as the amount of allocated and available storage. Information about the allocations is shown for the entire job, not only for Db2 Cloning Tool.

User response:

No action is required.

CKZ52002E Subtask *subtask_number*, Extent requested and SECQTY is 0, dsn *data_set_name*

Explanation:

During the log apply process, the target data set required an additional extent, but the object related to the data set was defined with SECQTY 0. The data set is not extended. Processing terminates.

User response:

Disable log apply processing, or create the target object with SECQTY -1 or *n* and start the cloning process from the beginning.

CKZ52500I IX-EXCLUDE *parm_name* = *parm_value*

Explanation:

This informational message lists the value of the IX_EXCLUDE command field.

User response:

No action is required.

CKZ52501E Invalid syntax in IX-EXCLUDE command, field *parm_name*

Explanation:

An error occurred during IX-EXCLUDE command parsing.

User response:

Contact IBM Software Support.

CKZ52502E Invalid value in IX-EXCLUDE command, field *parm_name*

Explanation:

An error occurred during IX-EXCLUDE command parsing.

User response:

Contact IBM Software Support.

CKZ52503E IX-EXCLUDE command parsing error, RC=*return_code*, RS=*reason_code*

Explanation:

An error occurred during IX-EXCLUDE command parsing.

User response:

Contact IBM Software Support.

CKZ52504E IX-EXCLUDE *parm_name* *parm_value* must be from *nn1* to *nn2* characters

Explanation:

The value of the parameter *parm_name* is the wrong length.

User response:

Correct the input and resubmit the job. If unable to resolve the error (for example, the error occurred during SYNCDB2 parsing), contact IBM Software Support.

CKZ52505E IX-EXCLUDE *parm_name* valid values are from *nn1* to *nn2*

Explanation:

The parameter *parm_name* is not valid. It must be in the range that is listed in the message.

User response:

Correct the input and resubmit the job. If unable to resolve the error (for example, the error occurred during SYNCDB2 parsing), contact IBM Software Support.

CKZ52601E SYNCDOCID command parsing error, RC=*return_code*, RS=*reason_code*

Explanation:

An error occurred during SYNCDOCID command parsing.

User response:

Contact IBM Software Support.

CKZ52602E Invalid syntax in SYNCDOCID command, field *field_name*

Explanation:

An error occurred during SYNCDOCID command parsing.

User response:

Contact IBM Software Support.

CKZ52603I SYNCDOCID *field_name* = *value*

Explanation:

This informational message lists the value of the SYNCDOCID command field.

User response:

No response is required.

CKZ52604E SYNCDOCID *parm_name* *parm_value* must be from *nn1* to *nn2* characters

Explanation:

The parameter *parm_name* value has an incorrect length.

User response:

Correct the input and resubmit the job. If unable to resolve this error, contact IBM Software Support.

CKZ52701I LOGAPPLY Command
command_name is string_value

Explanation:

This message displays the string value of the target input log apply command.

User response:

No action is required.

CKZ52702I LOGAPPLY Command
command_name is hex_value

Explanation:

This message displays the hex value of the target input log apply command.

User response:

No action is required.

CKZ52703I LOGAPPLY Command
command_name is decimal_value

Explanation:

This message displays the decimal value of the target input log apply command.

User response:

No action is required.

CKZ52704E LOGAPPLY Command Parsing
Error, RC=*return_code*,
RS=*reason_code*

Explanation:

The LOGAPPLY target input command cannot be parsed.

User response:

Correct the error and resubmit the job.

CKZ52705E LOGAPPLY *command_name*,
command_value Invalid Length

Explanation:

The LOGAPPLY target input command has an invalid length.

User response:

Correct the error and resubmit the job.

CKZ52707I LOGAPPLY Command Data Sharing
Member, ID=*data_sharing_ID*
SSID=*source_subsystem_for_this_ID*,
ZPARM=*ZPARM_member_for_this_ID*,
BSDS01=*BSDS01_data_set_for_this_ID*,
BSDS02=*BSDS02_data_set_for_this_ID*

Explanation:

The data sharing member information.

User response:

No action is required.

CKZ52708E LOGAPPLY *parameter_string*
invalid value, *value*

Explanation:

The LOGAPPLY parameter that is listed in the message has an invalid value.

User response:

Correct the error and resubmit the job.

CKZ52709W Specified GMT-OFFSET value is
invalid and will not be used. Valid
range is from -12:00 to +14:00

Explanation:

The GMT-OFFSET value that was entered is invalid. The GMT-OFFSET parameter is ignored.

User response:

Correct the invalid GMT-OFFSET.

CKZ52710W Invalid GMT-OFFSET format,
specified value will not be used.
Valid format is +HH:MM or -HH:MM

Explanation:

The GMT-OFFSET value that was entered is in an invalid format. The GMT-OFFSET parameter is ignored.

User response:

Enter the GMT-OFFSET in a valid format.

CKZ52711W GMT-OFFSET was specified with
END-POINT value other than
TO_TIMESTAMP or TO_TIMESTAMP
combined with USE-LOCAL-
TIME(Y), specified GMT-OFFSET
value will not be used

Explanation:

An invalid combination of parameters was entered. The GMT-OFFSET parameter will be ignored.

User response:

Specify END-POINT TO_TIMESTAMP and USE-LOCAL-TIME(N) to allow the GMT OFFSET value to be used.

CKZ52712E LOGAPPLY *command_name value*
must be *decimal_number*
characters maximum

Explanation:

The LOG-APPLY parameter has an invalid value. The parameter must be between 1 and *decimal_number* characters in length and cannot be null.

User response:

Correct the error and resubmit the job.

CKZ52801E TABLEDEF command parsing error,
RC=*return_code*, RS=*reason_code*

Explanation:

This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52802E **TABLEDEF**
TABLEDEF_field_name_in_error,
first_16_characters_of_field Length
exceeded

Explanation:

The field that is listed in the message contains a value that is too long. This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52803E **TABLEDEF**
TABLEDEF_field_name_in_error,
first_16_characters_of_field invalid
value

Explanation:

The field that is listed in the message contains an invalid value. This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52804E **TABLEDEF, null field,**
TABLEDEF_field_name_with_null_v
alue

Explanation:

The field that is listed in the message contains a null value. This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52805E **TABLEDEF, field length exceeded,**
TABLEDEF_field_name_with_length
_error

Explanation:

The field that is listed in the message contains a value that is too long. This error might be a user error if the

SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52806E **TABLEDEF, runaway field,**
field_with_no_detectable_end

Explanation:

The field that is listed in the message has no detectable end. This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52807E **TABLEDEF, premature end,**
field_with_premature_end

Explanation:

The field that is listed in the message ends prematurely. This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52808I **TABLEDEF field_name is =**
decimal_value

Explanation:

This message is informational.

User response:

No action is required.

CKZ52809I **TABLEDEF field_name is =**
string_value

Explanation:

This message is informational.

User response:

No action is required.

CKZ52810I **TABLEDEF field_name is =**
hex_value

Explanation:

This message is informational.

User response:

No action is required.

CKZ52901E **MASKCMD command parsing error,**
RC=return_code, RS=reason_code

Explanation:

This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52902E **MASKCMD**
MASKCMD_field_name_in_error,
first_16_characters_of_field length
exceeded

Explanation:

The field that is listed in the message contains a value that is too long. This error might be a user error if the SYNCDB2 data set has been edited, or the file might be corrupted.

User response:

Correct the error. This might require re-running the source job that builds the data set member. Resubmit the job.

CKZ52909I **MASKCMD field_name is =**
string_value

Explanation:

This message is informational.

User response:

No action is required.

CKZ53001I **BEGIN COMMAND FILE SYNTAX**
CHECKING

Explanation:

This message is informational.

User response:

No action is required.

CKZ53002W **parm_name parm has an invalid**
value, parm_value, defaulting to
default_string_value

Explanation:

This is a user error. The parameter *parm_name* has an invalid value.

User response:

Correct the input and resubmit the job, or use the default value.

CKZ53003I **Parmlib parm_name set to**
parm_value

Explanation:

This message is informational.

User response:

No action is required.

CKZ53004I **parm_name parm not in PARMLIB,**
defaulting to default_string

Explanation:

This message is informational.

User response:

No action is required.

CKZ53005W **parm1 is enabled, parm2 is**
disabled

Explanation:

Because *parm1* was enabled, *parm2* was automatically disabled.

User response:

Correct the input and resubmit the job, or use the values that were automatically set.

CKZ53009E **PGM(program) was specified,**
command command disabled

Explanation:

An invalid command was specified with PGM(SRCIMCPY) or PGM(SRCVSCPYPY). The following commands are not valid with PGM(SRCIMCPY) or PGM(SRCVSCPYPY): SIM(A), DATA-MASKING, and SUBTASK-DATASET-EXTENSIONS.

User response:

Change the invalid command to the default and resubmit the job.

CKZ53080E **mmm SUPPORT MODULE MISSING**

Explanation:

This is an error WTO. mmm = module name

User response:

Ensure the product is installed correctly and STEPLIB points to the correct CKZ libraries.

CKZ53081E **dd DD STATEMENT MISSING**

Explanation:

This is an error WTO. dd = the missing DD card

User response:

Add the DD to the JCL and resubmit the job.

CKZ53083E **MUST EXECUTE AS A Z/OS**
AUTHORIZED PROGRAM

Explanation:

This is a user error.

User response:

Authorize the CKZLOADLIB and resubmit the job.

CKZ53084E **UNABLE TO LOAD PROGRAM: ppp**

Explanation:

This is a user error. ppp = module unable to load

User response:

After successful installation of CKZ, resubmit the job.

CKZ53091E **FATAL ERRORS HAVE OCCURRED**
DURING //CKZINI PROCESSING

Explanation:

This is probably a user error.

User response:

Correct the error in Parmlib and resubmit the job.

CKZ53201I *parm_name* parm not in PARMLIB, defaulting to *default_decimal*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53202E DD NAME *ddname* IN *fff* NOT FOUND IN JCL

Explanation:

This is a user error. *ddname* = DD name not found in the JCL *fff* = COPY parm with the bad DD name

User response:

Correct the JCL or the COPY Command DD Name and resubmit the source job.

CKZ53203W *parm_name* parm has an invalid value, *value*, defaulting to *default_decimal*

Explanation:

An invalid parameter value was entered.

User response:

Correct the parameter value in the PARMLIB and resubmit the source job.

CKZ53204W *parm_name* parm has an invalid value, *value*, defaulting to *default_decimal*

Explanation:

An invalid parameter value was entered.

User response:

Correct the input and resubmit the job or use the default value.

CKZ53205I Parmlib *parm_name* set to *decimal_value*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53206W *parm_name* parm has an invalid value, *parm_value*, defaulting to *default_string_value*

Explanation:

An invalid parameter value was entered.

User response:

Correct the input and resubmit the job or use the default value.

CKZ53207I PARMLIB *parm_name* set to *parm_value*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53219I *parm_name* parm not in PARMLIB, defaulting to *decimal_value* seconds

Explanation:

This message is informational.

User response:

No action is required.

CKZ53220W *parm_name* parm has an invalid value, *decimal_value*, defaulting to *decimal_value* seconds

Explanation:

An invalid parameter value was specified.

User response:

Correct the input and resubmit the job or use the default value.

CKZ53223I PARMLIB *parm_name* set to *decimal_value* seconds

Explanation:

This message is informational.

User response:

No action is required.

CKZ53224I *parm_name* parm not in PARMLIB, defaulting to *default_string*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53227I COPY_OPTION *ccc* Not in Parmlib, Defaulting to *sss*

Explanation:

This message is informational. *ccc* = COPY_OPTION string *sss* = Y or N default value

User response:

No action is required.

CKZ53228W COPY_OPTION *option_name* has an invalid value, *value*, defaulting to *default_value*

Explanation:

An invalid value was entered for the COPY_OPTION parameter that is listed in the message.

User response:

Correct the input and resubmit the job or use the default value.

CKZ53229I **Parmlib COPY_OPTION ccc set to vvv**

Explanation:

This message is informational. ccc = COPY_OPTION string vvv = COPY_OPTION value

User response:

No action is required.

CKZ53236I **TCPIP_KEY_LABEL parm not in PARMLIB, will use default key**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53237W **TCPIP_KEY_LABEL parm has an invalid value, value, will use default key**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job, or use the default encryption key.

CKZ53380E **module_name support module missing**

Explanation:

An error occurred during module loading.

User response:

Contact IBM Software Support.

CKZ53381E **ddname DD statement missing**

Explanation:

The DD that is listed in the message is missing from the JCL.

User response:

Correct the input and resubmit the job.

CKZ53401I **Data masking definition, field_name = string_value**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53402E **MASKDEF command parsing error, RC=return_code, RS=reason_code**

Explanation:

An error was encountered in the MASKDEF command. The command syntax might have been entered incorrectly.

User response:

Correct the error and resubmit the job.

CKZ53403E **Required keyword, keyword missing from command**

Explanation:

An error was encountered in the MASKDEF command. The required keyword that is listed in the message is missing from command syntax.

User response:

Add the missing keyword and resubmit the job.

CKZ53404E **MASKDEF_field_name, string_value, invalid length**

Explanation:

An error was encountered in the MASKDEF command. The field value that is listed in the message is too long.

User response:

Correct the error and resubmit the job.

CKZ53405E **No TABLECREATOR specified and no default creator specified using the SET command**

Explanation:

The table creator is missing from the MASKDEF command, and no default table creator has been specified via the SET command. A table creator name is required.

User response:

Either specify the TABLECREATOR in the MASKDEF command, or specify a DEFAULT-CREATOR using the SET command, and resubmit the job.

CKZ53500E **COPY COMMAND PARSING ERROR, RC=return_code, RS=reason_code**

Explanation:

An error has been returned from the initialization parsing routine. This is a probable user error.

User response:

Correct the input and resubmit the job.

CKZ53501E **COPY parm_name, invalid value parm_value**

Explanation:

An invalid parameter value was entered.

User response:

Correct the input and resubmit the job.

CKZ53502I **COPY TARGET, SSID sss, LOCATION lll, USERID uuu, PASSWORD ppp**

Explanation

This message is informational.

- sss = target DB2 subsystem
- lll = target DB2 subsystem location
- uuu = user id for DDF
- ppp = password for DDF (asterisks)

User response:

No action is required.

CKZ53503I **COPY *parm_name* is
*parm_string_value***

Explanation:

This message is informational.

User response:

No action is required.

CKZ53504I **SERVER IPV4 IS IPADDR**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53505I **COPY *parm_name* is
*parm_decimal_value***

Explanation:

This message is informational.

User response:

No action is required.

CKZ53506E **COPY *parm_name*, value
parm_value, invalid length**

Explanation:

An invalid parameter value was entered.

User response:

Correct the input and resubmit the job.

CKZ53507E **COPY *parm_name parm_value*
must be from *dd1* to *dd2*
characters**

Explanation:

The parameter *parm_name* value is not in the allowable range.

User response:

Correct the input and resubmit the job.

CKZ53508E **COPY *parm_name*, one or more
duplicates**

Explanation:

Duplicate values were specified for the parameter.

User response:

Correct the input and resubmit the job.

CKZ53509E **COPY LOG-APPLY *parm_name*,
value *parm_value*, invalid length**

Explanation:

An invalid parameter value was entered.

User response:

Correct the input and resubmit the job.

CKZ53516I **OBJECT TRANSLATE,
SOURCE=*sss*, TARGET=*ttt***

Explanation:

This message is informational. *sss* = source DB2 creator *ttt* = target DB2 creator

User response:

No action is required.

CKZ53518W **OBJECT TRANSLATE, DUPLICATE
OBJECT, *object_type*, SOURCE
*object_name***

Explanation:

This is a user error. The object duplicates the object name in a previous OBJXLATE command. This will result in duplicates not being used and therefore objects may not match for the copy.

User response:

Correct the input and resubmit the job. This may cause multiple jobs to be submitted with updated LISTDEF cards to eliminate the duplicates.

CKZ53536E **SIMULATE must be empty, blank,
Y, N, or A**

Explanation:

An invalid value was provided for SIMULATE.

User response:

Correct the input and resubmit the job.

CKZ53537E **INVALID SERVER IP *ipaddr*,
REASON *rrr***

Explanation

This is a user error. *ipaddr* may be IPv4 or IPv6.

rrr = reason code:

- 1** character not decimal digit or a dot
- 2** too many digits with no dot
- 3** no digits for octet
- 4** number of octets not 4
- 11** character not hex digit or colon
- 12** too many digits with no colon
- 13** too many hex pieces

- 14 more than 1 :: or :::
- 15 string > 39 characters
- 16 starts or ends with a single colon
- >20 contact IBM Software Support

User response:
Correct the input and resubmit the job or if >20, contact IBM Software Support.

CKZ53538I EXTEND-TARGET-PBG-TABLESPACE is obsolete and will not be supported in the future. Target PBG table spaces will be extended automatically

Explanation:
This message is informational, but JCL changes are required because EXTEND-TARGET-PBG-TABLESPACE parameter is obsolete.

User response:
Correct the JCL before you resubmit the job.

CKZ53539E FAST-REP OPERAND vvv IS INVALID

Explanation:
This is a user error. vvv = operand string

User response:
Correct the input and resubmit the job.

CKZ53540E UNRECOGNIZED KEYWORD IN OBJECT-TRANSLATION COMMAND ccc

Explanation:
This is a user error. ccc = command value

User response:
Correct the input and resubmit the job.

CKZ53541E ILLEGAL SYNTAX IN OBJECT-TRANSLATION COMMAND ccc

Explanation:
This is a user error. ccc = command value

User response:
Correct the input and resubmit the job.

CKZ53542E INVALID SERVER IP ipaddr, UNABLE TO DETERMINE IF V4 OR V6 FORMAT

Explanation:
This is a user error. ipaddr may be IPv4 or IPv6.

User response:
No action is required.

CKZ53543I SERVER IPV6 IS ipaddr

Explanation:
This message is informational. ipaddr = IPv6 address

User response:
No action is required.

CKZ53549W DSNS-PER-COPY is ddd, Out of Range (1-255), Reverting to Parmlib or Default Value

Explanation:
This message is a warning. ddd = decimal value entered

User response:
Correct the input and resubmit the job if MAX_RC=0. If MAX_RC=4, this warning will be ignored.

CKZ53550W DSS-COPY-COMMANDS is ddd, Out of Range (1-256), Reverting to Parmlib or Default Value

Explanation:
This message is a warning. ddd = decimal value entered

User response:
Correct the input and resubmit the job if MAX_RC=0. If MAX_RC=4, this warning will be ignored.

CKZ53551W AUTO-START-SOURCE-SPACE(Y) or AUTO-START-SOURCE-SPACE(R), and AUTO-START-TARGET-SPACE(Y) should be used to avoid issues with UNLOAD/LOAD utilities

Explanation:
To avoid problems with the UNLOAD/LOAD utilities, it is recommended to specify AUTO-START-SOURCE-SPACE(Y) or AUTO-START-SOURCE-SPACE(R), and AUTO-START-TARGET-SPACE(Y) with UNLOAD-LOAD-ENABLE(Y).

User response:
Correct the input and resubmit the job.

CKZ53552W AUTO-START-TARGET-SPACE should be set to "Y" to avoid issues with REBUILD utility

Explanation:
To avoid problems with the REBUILD INDEX utility, it is recommended to specify AUTO-START-TARGET-SPACE(Y) with REBUILD-INDEXES-EXECUTE (Y).

User response:
Correct the input and resubmit the job.

CKZ53553E COPY DATA-MOVER CMDDNAME ddname is invalid

Explanation:

An invalid parameter was entered. The length of the ddname that is listed in the message is incorrect.

User response:

Correct the input and resubmit the job.

**CKZ53554E JOB-TEMPLATE DD ddname IS
 INVALID**

Explanation:

This is a user error. The length of the DD name is invalid.

User response:

Correct the input and resubmit the job.

**CKZ53555I JOB-TEMPLATE DD PAIR,
 INDD=ddname, OUTDD=ddname**

Explanation:

This message is informational. One message prints for each pair in the JOB-TEMPLATE subcommand of the COPY command.

User response:

No action is required.

**CKZ53558E ALWAYS-COPY-INDEXSPACES
 must be "Y" if parameter
 RECREATE-OBJECTS is A or M**

Explanation:

When RECREATE-OBJECTS is set to A or M, ALWAYS-COPY-INDEXSPACES must be set to Y.

User response:

Correct the input and resubmit the job.

**CKZ53559E INCLUDE-ALL-RI must be "Y" if
 parameter RECREATE-OBJECTS is
 A or M**

Explanation:

When RECREATE-OBJECTS is set to A or M, INCLUDE-ALL-RI must be set to Y.

User response:

Correct the input and resubmit the job.

**CKZ53560E COPY DDL PROCESS-TYPE must
 be N, Y, G, X or A**

Explanation:

The value that is specified for the PROCESS-TYPE parameter is not allowed. Allowed values are N, Y, G, X, or A.

User response:

Correct the error and resubmit the job.

**CKZ53561E RECREATE-OBJECTS must be A, M
 or N**

Explanation:

The value that is specified for the RECREATE-OBJECTS parameter is not allowed. Allowed values are A, M, or N.

User response:

Correct the error and resubmit the job.

**CKZ53562W Parameter RECREATE-
 OBJECTS(*parm_value*) is not valid
 for PROCESS-TYPE A, X or N.
 Defaulting to N**

Explanation:

The specified value for the RECREATE-OBJECTS parameter is not valid for the specified value of the PROCESS-TYPE parameter. The RECREATE-OBJECTS parameter will be set to "N".

User response:

If you want to use the re-creation function for target objects, set the PROCESS-TYPE value to Y or G and resubmit the job.

**CKZ53563E DDL-ATTRIBUTE-CHANGE,
 Duplicate Card, *attribute_name*,
 source_string_value,
 target_string_value,
 object_change_applies_to,
 *source_mask***

Explanation:

This is an error.

User response:

Correct the error and resubmit the job.

**CKZ53565E Unrecognized Keyword in DDL-
 ATTRIBUTE-CHANGE Command,
 *command_name***

Explanation:

This is an error. The incorrect keyword is listed in the message.

User response:

Correct the error and resubmit the job.

**CKZ53566I DDL-ATTRIBUTE-CHANGE,
 Attribute=*attribute_name*, Source
 Value=*source_string_value*, Target
 Value=*target_string_value*,
 Applytoobj=*applytoobject*,
 Applytomask=*source_mask***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ53567E Illegal Syntax in DDL-ATTRIBUTE-
 CHANGE Command,
 *command_string***

Explanation:

This is an error.

User response:

Correct the error and resubmit the job.

CKZ53568E Invalid Value for GBPCACHE, Must be CHANGED, ALL, SYSTEM, or NONE

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53569E Invalid Value for LOG, Must be YES or NO

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53570E Invalid Value for TRACKMOD, Must be YES or NO

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53571E Invalid Value for CLOSE, Must be YES or NO

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53572E Invalid Value for DATACAPTURE, Must be NONE or CHANGES

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53573E Applytotype *applytotype* is not valid for Statement *statement*

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53576E DDL-ATTRIBUTE-CHANGE, Source Change Equals Target Change, Attribute=*attribute*, Src=*source_attribute_value*, Trg=*target_attribute_value*

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53577E Applytotype *applytotype_object_type* is not valid for Attribute *attribute* using Value *source_attribute_value*

Explanation:
This is an error.

User response:
Correct the error and resubmit the job.

CKZ53578I COPY LOG-APPLY Command *command_name* is *string_value*

Explanation:
The string value of the target input log apply command.

User response:
No action is required.

CKZ53579I COPY LOG-APPLY Command *command_name* is *decimal_value*

Explanation:
The decimal value of the target input log apply command.

User response:
No action is required.

CKZ53580E *command_name value* must be *decimal_number* Characters Maximum

Explanation:
The LOG-APPLY parm has an invalid value. The parm must be between 1 and *decimal_number* characters in length and cannot be null.

User response:
Correct the error and resubmit the job.

CKZ53584E Invalid BUFFERPOOL Pairing, Source Pool=*source_buffer_pool*, Target Pool=*target_buffer_pool*

Explanation:
This is an error. The size of the page cannot be changed.

User response:
Correct the error and resubmit the job.

CKZ53585I *Source | Target* catalog prefetch databases *database1, database2, database3, database4, database5.*

Explanation:
This message is informational. Up to 5 database names may print with this message. Multiple messages will print if there are more than 5 databases.

User response:
No action is required.

CKZ53586E *ddname* Allocation Error,
RC=return_code, RS=reason_code,
Error
Code=dynamic_allocation_return_c
ode

Explanation:

The data set could not be allocated.

User response:

Correct the error and resubmit the job.

CKZ53587I **Data Sharing Member,**
ID=data_sharing_ID,
SSID=source_subsystem_for_this_I
D,
ZPARM=ZPARM_member_for_this_
ID

Explanation:

The data sharing member information.

User response:

No action is required.

CKZ53591E **END-POINT RBA/LRSN**
value_entered is invalid, code
return_code

Explanation

An incorrect value was entered for the RBA or LRSN. The *value_entered* displays the invalid value in EBCDIC. The *return_code* listed in the message identified the cause, and can be one of the following:

- 08: Opening "X" was not found.
- 12: Ending single quotation mark was not found.
- 16: The RBA length must be 6 or 10 bytes.
- 20: The RBA length is not 10 bytes when DSNJCNVT has been run.
- 24: The RBA has one or more invalid hexadecimal characters.

User response:

Correct the input and resubmit the job.

CKZ53592E **PGM(program) specified, LOG-**
APPLY command was not found or
is incomplete

Explanation:

This error indicates that the LA-ENABLE command is N or defaulted, or no ZPARM member was entered, or both.

User response:

Provide the correct LOG-APPLY parameters for the specified PGM and resubmit the job.

CKZ53593W **The END-POINT for the source job**
with PGM(program) changed to
TO_CURRENT

Explanation:

The only valid END-POINT for the PGM parameter that is listed in the message is TO_CURRENT.

User response:

Correct the input and resubmit the job.

CKZ53594I **Object mismatch set for**
mismatch_type, return_code.

Explanation:

This message is informational. It displays the object mismatch type and return code that was specified in the COPY command.

User response:

No action is required.

CKZ53595E **Illegal syntax in OBJECT-**
MISMATCH-RETURN-CODE
command.

Explanation:

An invalid parameter was entered in the OBJECT-MISMATCH-RETURN code command.

User response:

Specify a valid mismatch code and return code in the command and resubmit the job.

CKZ53597I **COPY DATA-MOVER parm_name is**
parameter value | blank

Explanation:

This message is informational.

User response:

No action is required.

CKZ53598E **If USE-QUIESCE-POINT-FOR-**
LOGPOINT(Y) and QUIESCE(Y),
END-POINT cannot be specified

Explanation:

An error on the LOG-APPLY command was encountered.

User response:

Remove the END-POINT parameter and resubmit the job.

CKZ53599W **LOAD-SORTNUM is sortnum_value,**
out of range (2-255), reverting to
default value

Explanation:

This message is a warning that the LOAD-SORTNUM value is invalid.

User response:

If MAX_RC=0, correct the input and resubmit the job. If MAX_RC=4, this warning will be ignored.

CKZ535102I **DDL Attribute Change, Attribute**
GBPCACHE with value SYSTEM

will be applied for LOB objects only

Explanation:

Value SYSTEM for attribute GBPCACHE can be specified only for LOB objects

User response:

No action is required.

CKZ535103I DSS-COPY-COMMANDS * DSNS-PER-COPY = *number* - may cause storage and other resource-related errors during program execution. The recommended maximum for DSS-COPY-COMMANDS * DSNS-PER-COPY is 1400.

Explanation:

The supplied values for DSS-COPY-COMMANDS and DSNS-PER-COPY multiplied together give a product of *number*. This might cause storage and other resource-related errors during execution of the program that is listed in the message. The recommended maximum for DSS-COPY-COMMANDS * DSNS-PER-COPY is 1400.

User response:

No action is required.

CKZ535104W Specified GMT-OFFSET value is invalid and will not be used. Valid range is from -12:00 to +14:00

Explanation:

The GMT-OFFSET value that was entered is invalid. The GMT-OFFSET parameter is ignored.

User response:

Correct the GMT-OFFSET value.

CKZ535105W Invalid GMT-OFFSET format, specified value will not be used. Valid format is +HH:MM or -HH:MM

Explanation:

The GMT-OFFSET value that was entered is in an invalid format. The GMT-OFFSET parameter is ignored.

User response:

Enter the GMT-OFFSET in a valid format.

CKZ535106W GMT-OFFSET was specified with END-POINT value other than TO_TIMESTAMP or TO_TIMESTAMP combined with USE-LOCAL-TIME(Y), specified GMT-OFFSET value will not be used

Explanation:

An invalid combination of parameters was entered. The GMT-OFFSET parameter will be ignored.

User response:

Specify END-POINT TO_TIMESTAMP and USE-LOCAL-TIME(N) to allow the GMT-OFFSET value to be used.

CKZ535107W Parameter SKIP-LOG-APPLY(Y) is valid only if PGM(SRCIMCPY) or PGM(SRCVSCPY) specified. Defaulting to N.

Explanation:

SKIP-LOG-APPLY(Y) can be used only with PGM(SRCIMCPY) or PGM(SRCVSCPY).

User response:

Correct the input and resubmit the job or use the default value.

CKZ535108W Parameter USE-LAST-CONSISTENT-FLASHCOPY(Y) cannot be used with DATA-MOVER-PGM(*parameter*) and will be turned off.

Explanation:

USE-LAST-CONSISTENT-FLASHCOPY is valid only with DATA-MOVER PGM(ADR DSSU), PGM(EMCAPI), or PGM(NONE).

User response:

Correct the input and resubmit the job.

CKZ535109W COPY parameter *parameter_name*(Y) is not valid with SKIP-LOG-APPLY(Y). Defaulting to N

Explanation:

The parameter value that is listed in the message is not valid with SKIP-LOG-APPLY(Y).

User response:

Correct the input and resubmit the job or use the default value.

CKZ535110W COPY parameter *parameter* has invalid value. Defaulting to *default_value*.

Explanation:

The value for the parameter that is listed in the message is not valid. The default value will be used instead.

User response:

If the default value for the parameter is acceptable, no action is required. Otherwise, correct the parameter value and resubmit the job.

CKZ535111E PROCESS-DDL-DDN is required unless PROCESS-TYPE is N.

Explanation:

PROCESS-DDL-DDN (*ddname*) was not supplied. This parameter must be provided unless PROCESS-TYPE is N.

User response:

Correct the parameter and resubmit the job.

CKZ535112W **Parameter *parameter* cannot be used with USE-LAST-CONSISTENT-FLASHCOPY(Y) and will be turned off.**

Explanation:

USE-LAST-CONSISTENT-FLASHCOPY(Y) is not valid with the parameter that is listed in the message. The parameter will be turned off.

User response:

Correct the parameter and resubmit the job.

CKZ535113I **COPY parameter CHECK-INDEX-KEYS = N is not valid with *parameter_name* (Y), set to Y**

Explanation:

The *parameter_name* (Y) parameter was set, but the CHECK-INDEX-KEYS parameter was set to N. CHECK-INDEX-KEYS(Y) is required for *parameter_name*(Y). Db2 Cloning Tool set the CHECK-INDEX-KEYS parameter to Y.

User response:

No action is required.

CKZ535115W **PGM(*program*) was specified with FUZZY-COPY(Y), but LOG-APPLY and USE-LAST-CONSISTENT-FLASHCOPY are disabled. That may cause data inconsistency on target. LOG-APPLY(Y) or USE-LAST-CONSISTENT-FLASHCOPY(Y) must be specified.**

Explanation:

To avoid data inconsistency on the target, when FUZZY-COPY is specified, and ADRDSSU, EMCAP1, or NONE is specified for PGM, LOG-APPLY(Y) or USE-LAST-CONSISTENT-FLASHCOPY(Y) must also be specified.

User response:

Correct the input and resubmit the job.

CKZ535117W ***Parm_name* is *parm_value*, out of range (*min_value*-*max_value*), reverting to default value**

Explanation:

The specified *parm_value* is invalid for the *parm_name*. The parameter's default value will be used.

User response:

If MAX_RC=0, correct the input and resubmit the job. If MAX_RC=4, this warning may be ignored.

CKZ535118E **The following DD name cannot be used explicitly as a Db2 Cloning Tool DD name: *ddname***

Explanation:

The specified *ddname* cannot be used explicitly in a Db2 Cloning Tool job.

User response:

Correct the input and resubmit the job.

CKZ535119E **Mask characters are not allowed for attribute *attribute_name* values in DDL-ATTRIBUTE-CHANGE-COMMAND, value *value***

Explanation:

Only STOGROUP, VCAT and BUFFERPOOL attributes values can contain mask characters.

User response:

If this error occurred for one of these parameters, contact IBM Software Support. Otherwise, correct the input and resubmit the job.

CKZ535120I ***parm1*(Y) is specified, *parm2* is set to Y**

Explanation:

Because *parm1* was set to Y, *parm2* was automatically set to Y.

User response:

No action is required.

CKZ535121I ***parm1* is enabled, *parm2* is disabled**

Explanation:

Because *parm1* was enabled, *parm2* was automatically disabled.

User response:

No action is required.

CKZ535122E **Value specified for *parm_name* is too big. Maximum value is *max_value***

Explanation:

An incorrect value was specified for the parameter that is listed in the message.

User response:

Correct the input and resubmit the job.

CKZ535123I ***option_name* is ignored because DATA-MOVER PGM value is not ADRDSSU**

Explanation:

The option that is listed in the message was specified with an incompatible DATA-MOVER PGM value.

User response:

Correct the input and resubmit the job, or you can ignore this informational message and continue with the specified DATA-MOVER copy program.

CKZ535124W DATA-MOVER REUSE(N) parameter cannot be specified when DATA-MOVER PGM(ADDRSSU) is used. DATA-MOVER REUSE set to Y

Explanation:

ADDRSSU always reuses existing allocations.

User response:

Correct the input and resubmit the job.

CKZ535125W DATA-MOVER NULLSTORCLAS parameter cannot be specified when DATA-MOVER STORCLAS is also specified. DATA-MOVER NULLSTORCLAS set to N

Explanation:

The STORCLAS and NULLSTORCLAS keywords are mutually exclusive.

User response:

Correct the input and resubmit the job.

CKZ535126W DATA-MOVER DATACLAS parameter cannot be specified when DATA-MOVER PGM(ADDRSSU) is used. DATA-MOVER DATACLAS will be ignored

Explanation:

ADDRSSU does not support specifying DATACLAS for copied data sets. The supplied DATACLAS value will be ignored.

User response:

Correct the input and resubmit the job.

CKZ535127W *parm_name* value *parm_value* is invalid, defaulting to *default_decimal*

Explanation:

This message indicates that *parm_value* is not a valid value for the *parm_name* keyword. The default value will be used.

User response:

If MAX_RC=0, correct the input and resubmit the job. If MAX_RC=4, this warning may be ignored.

CKZ535128W FUZZY-COPY(Y) parameter is not compatible with DATA-MOVER PGM(SRCVSCP)

Explanation:

This message indicates that using DATA-MOVER PGM(SRCVSCP) and FUZZY-COPY(Y) may cause data inconsistency on the target.

User response:

Correct the input and resubmit the job.

CKZ535129W SKIP-LOG-APPLY(Y) will be used for DATA-MOVER PGM(SRCVSCP)

Explanation:

When using DATA-MOVER PGM(SRCVSCP) and FUZZY-COPY(Y), SKIP-LOG-APPLY (Y) is required and was automatically set.

User response:

No action is required.

CKZ535130E DATA-MOVER FCTOXRCPPRIMARY parameter is not compatible with DATA-MOVER FASTREP(NONE)

Explanation:

The DATA-MOVER FASTREP parameter must be PREF or REQ to use FCTOXRCPPRIMARY.

User response:

Correct the input and resubmit the job.

CKZ535131E Arguments are not expected for DATA-MOVER FCTOXRCPPRIMARY parameter

Explanation:

The FCTOXRCPPRIMARY parameter must be specified without arguments.

User response:

Correct the input and resubmit the job.

CKZ535133I REBUILD-COPY-NO-INDEXES is obsolete and will not be supported in the future. All indexes that are not included in image copy will be rebuilt.

Explanation:

This message is informational. When DATA-MOVER PGM(SRCIMCPY) is used, the rebuild of COPY NO indexes will be automatically completed, either in the target job when REBUILD-INDEXES-EXECUTE(Y) is specified in the SET command of the source job, or in a generated REBUILD INDEX job when either the CKZINTRB DD is specified in the target job JCL or the SET command parameter TARGET-JOB-INDEX-REBUILD-DDN is specified in the source job.

User response:

No action is required.

CKZ535134I REPLACE-TARGET-DSN is obsolete and will not be supported in the future. The target VSAM object will always be overwritten if it exists

Explanation:

This message is informational. The source job will automatically replace existing target VSAM data sets when cloning the data sets in non-simulate mode.

User response:

No action is required.

CKZ535135I **ALTER-FOR-XML-LOB-COLUMNS is obsolete and will not be supported in the future. ALTER statements will be used to create XML DOCID and LOB ROWID columns in the right order. AREO* and AREOR will be automatically skipped if DDL-ENABLE(Y).**

Explanation

This message is informational. DDL generation will use one or more ALTER commands in the target DDL when one (or more) of the following Db2-generated column types is not at the physical end of the source table row:

- DB2_GENERATED_DOCID_FOR_XML
- DB2_GENERATED_ROWID_FOR_LOBSnn

Target table spaces that match the above description might be in Advisory Reorg Pending (AREO* or AREOR) status after the clone. This status does not prevent data access; it is advisory only. You may leave the status as it is, run a REORG at your convenience, or use START ACCESS(FORCE) to clear the status. Specifying DDL-ENABLE(Y) changes the behavior of ADVISORY-STATUS-VALUES; when using the default values for that keyword, AREO* and AREOR are not treated as errors because AREO* and AREOR are expected in that situation. To treat them as errors, specify ADVISORY-STATUS-VALUES and explicitly include AREO* or AREOR in the list of values.

User response:

No action is required.

CKZ53600E **SET command parsing error, RC=return_code, RS=reason_code**

Explanation:

An error was returned from the initialization parsing routine. This is probably a user error.

User response:

Correct the command input and resubmit the job.

CKZ53601E **SET ERROR, TARGET-JOB AND TCP-SERVER-JOB CANNOT BOTH BE Y**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ53602I *parameter* **Parm Not in SET Command, Defaulting to number_of_seconds Seconds**

Explanation:

This message is informational and indicates that the default number of seconds will be used.

User response:

No action is required.

CKZ53604W *command_name* **SET COMMAND HAS AN INVALID VALUE, command_value, IGNORED**

Explanation:

This message is a warning and indicates that a value that is entered for the SET command is not valid. The product default or the PARMLIB value will be used.

User response:

Correct the input and resubmit the job if MAX_RC=0. If MAX_RC=4, this warning will be ignored.

CKZ53605E **Runtime repository dd_name DD is missing**

Explanation:

The SET command USE-RUNTIME-REPOSITORY(Y) is present and one or both DDs that are required for using the runtime repository are missing.

User response:

Ensure that both DD cards are available in the JCL stream or turn USE-RUNTIME-REPOSITORY off (N).

CKZ53606E **SET parm_name, value parm_value, invalid length**

Explanation:

This is a user error.

User response:

Correct the command input and resubmit the job.

CKZ53607E **Report job: No repository DDs found**

Explanation:

Neither of the input DDs CKZRRJOB or CKZRRDSN were found.

User response:

Correct the error and resubmit the job.

CKZ53608I **SET ttt-STATUS-VALUES USING DEFAULT STATUS VALUES**

Explanation:

This message is informational. ttt= ADVISORY or RESTRICT

User response:

No action is required.

CKZ53609I **No Default SQLID, MASKDEF rules must have TABLECREATOR specified**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53610E **SET *parm_name parm_value* must be from *dd1* to *dd2* characters**

Explanation:

An invalid value was entered for the parameter that is listed in the message.

User response:

Correct the input and resubmit the job.

CKZ53611E **ILLEGAL SYNTAX IN SET DEFAULT-SQLID COMMAND, *vvv***

Explanation:

This is a user error. *vvv* = command value

User response:

Correct the input and resubmit the job.

CKZ53612I **SET COMMAND *ccc* = *vvv***

Explanation:

This message is informational. *ccc* = command name
vvv = command value

User response:

No action is required.

CKZ53613E **ILLEGAL VALUE IN SET *ttt*-STATUS-VALUES, *sss***

Explanation:

This is a user error. *ttt* = ADVISORY or RESTRICT *sss* = bad status value

User response:

Correct the input and resubmit the job. Refer to the Command Reference for your version of DB2 for a list of valid values for the ?DISPLAY DATABASE command.

CKZ53614I **SET *ttt*-STATUS-VALUES USING STATUS VALUE *sss***

Explanation:

This message is informational. One message will print for each status value requested. *ttt* = ADVISORY or RESTRICT *sss* = bad status value

User response:

No action is required.

CKZ53615I **NO *ttt* STATUS COMMAND ISSUED PER PARM**

Explanation:

This message is informational. *ttt* = ADVISORY or RESTRICT

User response:

No action is required.

CKZ53616E ***sss* IS NOT A RECOGNIZED STATUS VALUE FOR *ttt*-STATUS-VALUES**

Explanation:

This is a user error. *sss* = bad status value *ttt* = ADVISORY or RESTRICT

User response:

Correct the input and resubmit the job. Refer to the Command Reference for your version of DB2 for a list of valid values for the ?DISPLAY DATABASE command.

CKZ53617W ***ttt*-STATUS-VALUES HAS A DUPLICATE STATUS VALUE FOR *sss***

Explanation:

This is a warning. The duplicate will be ignored if the job runs (MAX-RC=4). *ttt* = ADVISORY or RESTRICT *sss* = duplicate status value

User response:

Correct the input and resubmit the job.

CKZ53618E **ERROR PROCESSING STATUS VALUES**

Explanation:

This is a user error. See more detailed messages to determine the exact cause.

User response:

Correct the input and resubmit the job.

CKZ53619E **Report job: No report output DDs found**

Explanation:

None of the four output DDs (CKZJREPL, CKZJREPS, CKZDREPL, or CKZDREPS) were found.

User response:

Correct the error and resubmit the job.

CKZ53620I **SET TEMPLATE-VARIABLE, *name=variable_name*, *value=string_value_of_variable***

Explanation:

This message is informational. One message is generated for each variable that is specified.

User response:

No action is required.

CKZ53621E **TEMPLATE-VARIABLE *name variable_name* is invalid**

Explanation:

The length of the template variable name is invalid.

User response:

Correct the input and resubmit the job.

CKZ53622E **TEMPLATE-VARIABLE *name variable_name* matches a processing variable**

Explanation:

The variable name that is listed in the message specified matches the name of an internally defined variable.

User response:

Change the variable name and resubmit the job.

CKZ53623W **ccc SET COMMAND HAS AN INVALID VALUE, vvv**

Explanation:

This is a user error indicating a SET command has an invalid value. ccc = the name of the SET command vvv = the invalid value

User response:

Correct the SET command value and resubmit the job.

CKZ53624W **SET Command *parm_name* has an invalid value, *value*, defaulting to N**

Explanation:

This message is a warning that a value that is entered for the SET command is not valid. The value will be changed to N.

User response:

If MAX_RC=0, correct the input and resubmit the job. If MAX_RC=4, this warning will be ignored.

CKZ53625E **ddname must be from 1-7 characters**

Explanation:

The input or output job template member name has an invalid length.

User response:

Change the ddname and resubmit the job.

CKZ53626I **SET command source server IPv4 is *ip_address*.**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53627I **SET command source server IPv6 is *ip_address*.**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53628E **SET error, *parameter_1* and *parameter_2* are incompatible**

Explanation:

This is a user error. Specify one or the other of these parameters, but not both.

User response:

Correct the input and resubmit the job.

CKZ53629I **SET command server IPv4 is *ipaddr***

Explanation:

This message is informational.

User response:

No action is required.

CKZ53630I **SET command server IPv6 is *ip6addr***

Explanation:

This message is informational.

User response:

No action is required.

CKZ53631E ***server_job_1* SET command(s) found in the *server_job_2* TCPIP job**

Explanation

This is a user error. Source server commands cannot be in the target server and vice-versa.

- *server_job_1* = the server job that the SET commands belong to
- *server_job_2* = the server job in which the SET commands were found

User response:

Correct the input and resubmit the job.

CKZ53632E ***parameter value* must be *nnn* characters maximum**

Explanation:

The SET parameter that is listed in the message has an invalid value. The parameter must be between 1 and *nnn* characters in length and cannot be null.

User response:

Correct the error and resubmit the job.

CKZ53633I **SET command *command_name* = *command_value_in_decimal***

Explanation:

This message is informational.

User response:

No action is required.

CKZ53634W **Report job: *job* | *data set* report requested, but no ddname DD found**

Explanation:

This message is a warning. The report requested requires a DD card for input that was not present.

User response:

Correct the error and resubmit the job.

CKZ53635E Invalid server IP *ip_address*, reason *reason_code*

Explanation

This is a user error. Possible reason codes are:

- 1: Character is not a decimal digit or a period.
- 2: Too many digits with no period.
- 3: No digits for octet.
- 4: The number of octets is not 4.
- 11: A character is not a hex digit or colon.
- 12: Too many digits without a colon.
- 13: Too many hex digits.
- 14: More than 1 double colon (::) or triple colon (:::).
- 15: String is greater than 39 characters.
- 16: The string starts or ends with a single colon.
- >20: Contact IBM Software Support.

User response:

Correct the input and resubmit the job. If the reason code is greater than 20, contact IBM Software Support.

CKZ53636E Invalid Server IP *ip_address*, unable to determine if V4 or V6 format

Explanation:

This is a user error. *ip_address* may be IPv4 or IPv6.

User response:

Verify the IP value, correct the input, and resubmit the job. If the IP format is correct, contact IBM Software Support.

CKZ53637W REBUILD-UNMATCHED-TARGET-INDEXES(Y) is specified, PROCESS-UNMATCHED-TARGET-INDEXES defaulting to Y

Explanation:

Rebuild of unmatched target indexes cannot be executed without processing them.

User response:

No action is required.

CKZ53638W *parameter* (Y) is specified, but REBUILD-INDEXES-EXECUTE is set to N. *parameter* defaulting to N

Explanation:

The parameter that is listed in the message is required for REBUILD-INDEXES-EXECUTE(Y).

User response:

Correct the input and resubmit the job.

CKZ53639I *parm_name* value is set to Y. This value will always be used despite specified value in JCL

Explanation:

This message is informational. The value Y of the parameter *parm_name* will always be set to Y, regardless of the value that was specified in the submitted JCL.

User response:

No action is required.

CKZ53640W REBUILD-INDEXES-EXECUTE(Y) is specified, PROCESS-UNMATCHED-TARGET-INDEXES(Y) is also specified, REBUILD-UNMATCHED-TARGET-INDEXES defaulting to Y

Explanation:

The REBUILD for unmatched indexes will always be performed if unmatched indexes are processed and REBUILD-INDEXES-EXECUTE is enabled.

User response:

Correct the input and resubmit the job.

CKZ53641E Template variable *variable_name*, *variable_value* length exceeded

Explanation:

An invalid value was entered for the variable *variable_name* in the parameter TEMPLATE-VARIABLES.

User response:

Correct the input and resubmit the job.

CKZ53642I REMOTE-CONNECT-TYPE(D) is no longer supported. REMOTE-CONNECT-TYPE defaulting to "T"

Explanation:

This message is informational and indicates that a remote connection will be established via TCP/IP instead of DDF. DDF is no longer supported.

User response:

No action is required.

CKZ53643I TARGET-JOB-REPAIR-TEST is obsolete and will not be supported in the future.

Explanation:

This message is informational. The TARGET-JOB-REPAIR-TEST specified in the source job is ignored because it is not supported.

User response:

No action is required.

CKZ53644I TARGET-JOB-REPAIR-SELECT is obsolete and will not be supported in the future. REPAIR processing

will be automatically enabled if TARGET-JOB-REPAIR-EXECUTE(Y) is used or TARGET-JOB-REPAIR-DDN is set.

Explanation:

This message is informational. The REPAIR utility is automatically used to resolve mismatches if TARGET-JOB-REPAIR-EXECUTE(Y) is specified or TARGET-JOB-REPAIR-DDN is set.

User response:

No action is required.

CKZ53702E SYNCDB2 COMMAND PARSING ERROR, RC=rrr, RS=sss

Explanation:

Target job initialization has failed to read the SYNCDB2 input file using DD CKZIN. rrr = return code sss = reason code

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53704E MAXIMUM DATASET EXTENSIONS vvv, INVALID LENGTH

Explanation:

The MAXIMUM-EXTENSION value in the SYNCDB2 Command is invalid. vvv = invalid length

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53705E MAXIMUM DATASET EXTENSIONS HAS AN INVALID VALUE, vvv

Explanation:

The MAXIMUM-EXTENSION value in the SYNCDB2 Command is invalid. vvv = invalid value

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53709E TARGET-DSN dsname HAS A LENGTH >44

Explanation:

TARGET-DSN in the SYNCDB2 command has a length greater than 44.

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified

correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53710E CISIZE (NNN) MUST BETWEEN 4K AND 32K

Explanation:

The CISIZE is invalid.

User response:

Correct the CISIZE of the space and resubmit the source and target jobs.

CKZ53712E PAGE-SIZE (ddd) MUST BE 4, 8, 16 OR 32

Explanation:

PAGE-SIZE in the SYNCDB2 command has an invalid length. ddd = invalid value

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53716E STOP-SPACE (vvv) MUST BE Y OR N

Explanation:

STOP-SPACE in the SYNCDB2 command has an invalid value. vvv = invalid value

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53717E CI-HEADER (invalid_value) must be Y or N

Explanation:

CI-HEADER in the SYNCDB2 command has an invalid value. This error might be a user error if the SYNCDB2 data set has been edited.

User response:

If you edited the SYNCDB2 data set, correct the error and resubmit the job. If you have not edited the data set, ensure that the input file was specified correctly and is the same one that the corresponding source job output to the SYNCDB2-DDN.

CKZ53719E START-SPACE (vvv) MUST BE Y OR N

Explanation:

START-SPACE in the SYNCDB2 command has an invalid value. vvv = invalid value

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified

correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53720E *invalid_value, bad value
parm_name_in_error*

Explanation:

One of the following is invalid in the SYNCDB2 command: SELECT-FORMAT, SELECT-VERSION, SELECT-HASHPAGES or SELECT-RBA_FORMAT.

User response:

If you edited the SYNCDB2 data set, correct the error and resubmit the job. If you have not edited the data set, ensure that the input file was specified correctly and is the same one that the corresponding source job output to the SYNCDB2-DDN.

CKZ53729E **DUPLICATE ttt TRANSLATE
COMMANDS**

Explanation:

Duplicate translate commands found in a SYNCDB2 command. ttt = DBID | TSOB | PSID | ISOB | IXOB

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53730E **UNKNOWN IDTYPE ttt**

Explanation:

Unknown translate commands found in a SYNCDB2 command. ttt = unknown translate string

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53731E **sss OF ddd HAS EXCEEDED
MAXIMUM VALUE 65535**

Explanation:

The parm specified has an invalid value (> 16 bits). Object IDs are limited to 16 bits. sss = SRCPARM | TRGPARM ddd = value of the PARM

User response:

If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53736I **SYNCDB2 sss IS = ddd**

Explanation:

This message is informational. sss = parameter name ddd = decimal parameter value

User response:

No action is required.

CKZ53737I **SYNCDB2 parm_name is
parm_value**

Explanation:

This message is informational.

User response:

No action is required.

CKZ53738I **SYNCDB2 sss IS = vvv**

Explanation:

This message is informational. sss = parameter name vvv = parameter value in hex

User response:

No action is required.

CKZ53740I **SYNCDB2 parameter_name is =
parameter_value_in_decimal**

Explanation:

This message provides information about parameters.

User response:

No action is required.

CKZ53744I **SYNCDB2 sss IS = SRC(hhh)
TRG(hhh)**

Explanation:

This message is informational. sss = parameter name hhh = hex parameter value

User response:

No action is required.

CKZ53745E **SYNCDB2 object, parameter,
Invalid Length**

Explanation:

The database or indexespace in the SYNCDB2 member has an invalid length. *object* = TARGET-TS-DB or TARGET-TS-TS and *parameter* = value of the parameter.

User response:

If the file has been edited, this may be a user error; correct the parameter value. If not, ensure the input file was correctly specified and is the same file that the corresponding source job output to the SYNCDB2-DDN.

CKZ53746E **SYNCDB2 command_name value
must be from dd1 to dd2
characters maximum**

Explanation:

The value that was entered for the command that is listed in the message has an incorrect length.

User response:

Correct the input and resubmit the job.

CKZ53802E **HLQDDDF COMMAND PARSING
ERROR, RC=rrr, RS=sss**

Explanation:

Source job initialization has failed to read the HLQDDDF command. rrr = return code sss = reason code

User response:

Correct the input and resubmit the job.

CKZ53803E DIRECTION MUST BE IN OR OUT**Explanation:**

The HLQDDDF DIR parm must be IN or OUT.

User response:

Correct the input and resubmit the job.

CKZ53804E HLQNAME MUST BE 8 CHARACTERS OR LESS**Explanation:**

The HLQNAME value must be 8 characters or less. This is a data set naming restriction.

User response:

Correct the input and resubmit the job.

CKZ53805E HLQDDDF DUPLICATE DEFINITION FOR HLQ=vvv**Explanation:**

The HLQNAME value cannot be specified more than once. vvv = duplicate HLQNAME value

User response:

Correct the input and resubmit the job.

CKZ53806E Keywords *keyword1* and *keyword2* are mutually exclusive**Explanation:**

The keywords that are listed in the message cannot be specified with each other.

User response:

Correct the input and resubmit the job.

CKZ53807E REQUIRED KEYWORD, kkk MISSING FROM COMMAND**Explanation:**

The given keyword is required for the HLQDDDF command. kkk = missing keyword

User response:

Correct the input and resubmit the job.

CKZ53809E BAD FORMAT IN DDNAME vvv IN HLQDDDF COMMAND**Explanation:**

The format of the DDNAME is invalid. vvv = DDNAME with the bad format

User response:

Correct the input and resubmit the job.

CKZ53810E HLQNAME nnn DDNAME vvv NOT FOUND**Explanation:**

The DDNAME given on the HLQDDDF statement is not in the JCL. nnn = HLQ name this DD references vvv = DDNAME with the bad format

User response:

Correct the input and resubmit the job.

CKZ53811E BAD LENGTH IN DDNAME vvv IN HLQDDDF COMMAND**Explanation:**

The length of the DDNAME is invalid. vvv = DDNAME with the bad length

User response:

Correct the input and resubmit the job.

CKZ53813E HLQDDDF DD definition for HLQ=*hlq* has no definition specified**Explanation:**

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ53816E *parm* must be specified when DISP is not SHR or OLD**Explanation:**

The parameter that is listed in the message must be specified when DISP is not SHR or OLD.

User response:

Correct the input and resubmit the job.

CKZ53819E Bad length in VOLSER *volser* in HLQDDDF command**Explanation:**

This is a user error. The specified VOLSER is invalid.

User response:

Correct the input and resubmit the job.

CKZ53820E Bad format in VOLSER *volser* in HLQDDDF command**Explanation:**

This is a user error. The specified VOLSER is invalid.

User response:

Correct the input and resubmit the job.

CKZ53821E HLQDDDF DD definition for HLQ=*hlq* has too many volumes, use explicit DD definitions and DD name option**Explanation:**

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ53828I HLQDDDF HLQNAME is *hlq*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53829I HLQDDDF DD definition will be created, DISP=*disp*, UNIT=*unit*, VOLSERS (if any) to follow...

Explanation:

This message is informational.

User response:

No action is required.

CKZ53830I HLQDDDF DD definition VOLSER=*volser*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53831I HLQDDDF DD definition will be created, DISP=*disp*

Explanation:

This message is informational.

User response:

No action is required.

CKZ53832I HLQDDDF DD DEFINITION DDNAME=*ddname*, DIR=*ddd*

Explanation:

This message is informational. *ddd* = IN or OUT

User response:

No action is required.

CKZ53901E DATASUBTYPE Command Parsing Error, RC=*return_code*, RS=*reason_code*

Explanation:

Source job initialization has failed to read the DATASUBTYPE command.

User response:

Correct the input and resubmit the job.

CKZ53902E DATASUBTYPE *parm_name* *parm_value* must be from *dd1* to *dd2* characters

Explanation:

The DATASUBTYPE parameter *parm_name* has an incorrect length.

User response:

Correct the input and resubmit the job.

CKZ53903I DATASUBTYPE *field=value*

Explanation:

This gives the parsed value for the given field.

User response:

No action is required.

CKZ53904E Illegal Syntax in DATASUBTYPE Command, Field, *field_name*

Explanation:

The field has an invalid DB2 value.

User response:

Correct the input and resubmit the job.

CKZ53905E SUBTYPE must be S, B or M

Explanation:

The FOREIGNKEY value is incorrect.

User response:

Correct the input and resubmit the job.

CKZ54002W DATA MASKING REQUESTED FOR AT LEAST ONE TABLE IN DSN=*dsname*. HOWEVER NO OBJECT EXISTS ON THE TARGET, DATA MASKING DISABLED FOR THIS OBJECT

Explanation:

This message is a warning. *dsname* = data set name

User response:

Either change the MASKCMD or add the table definition on the target subsystem and resubmit the job.

CKZ54003W DATA MASKING REQUESTED FOR AT LEAST ONE TABLE IN DSN=*dsname*. HOWEVER THE DATASET COULD NOT BE COPIED, DATA MASKING DISABLED FOR THIS OBJECT

Explanation:

This message is a warning. *dsname*=data set name.

User response:

Either change the MASKCMD or correct the condition causing the data set not to copy and resubmit the job.

CKZ54004W DATA MASKING REQUESTED FOR TABLE *mmm.nnn*, COLUMN *ccc*, HOWEVER, THE COLUMN WAS NOT FOUND IN THE TABLE, ALL DATA MASKING IS NOW DISABLED

Explanation:

This message is a warning. Note that data masking for all columns is now disabled. *mmm* = table creator *nnn* = table name *ccc* = column name

User response:

Correct the column name in the MASKDEF command and resubmit the job.

CKZ54005I **DATA MASKING OKAY FOR TABLE**
mmm.nnn, COLUMN ccc

Explanation:

This message is informational. Data masking information will be passed to the target job to allow the masks to be applied to the data sets involved. mmm = table creator nnn = table name ccc = column name

User response:

No action is required.

CKZ54101E **ALTER statements are required for**
objects with identity columns, but
PROCESS-DDL-DDN for DDL
generation was not specified

Explanation:

PROCESS-DDL-DDN (*ddname*) was not supplied. This parameter must be provided if objects with identity columns are cloned.

User response:

Correct the parameter and resubmit the job.

CKZ54102E **ALTER statements are required for**
objects with identity columns, but
PROCESS-DDL block with
PROCESS-DDL-DDN was not
specified

Explanation:

The PROCESS-DDL block with PROCESS-DDL-DDN (*ddname*) was not supplied. This block and parameter must be provided if objects with identity columns are cloned.

User response:

Correct the parameter and resubmit the job.

CKZ54124I **Object *object_type***
object_qualifier_1.object_qualifier_
***2*, already exists on the target, No**
DDL create will be generated

Explanation:

This object already exists on the target. For a database, only one qualifier will be displayed in this message.

User response:

No action is required.

CKZ54126E **Subtask *subtask_number*, unable**
to process DSNHDECP, reason =
reason_code

Explanation:

An error occurred during DSNHDECP processing.

User response:

Contact IBM Software Support.

CKZ54200W **All COPY requests completed with**
errors, TSok *num_of_table_spaces*,
ISok *num_of_index_spaces*, TSerr
***num_of_table_spaces_with_errors*,**
ISerr
num_of_index_spaces_with_errors

Explanation:

This message is a warning because MAX-COPY-RC=8. If MAX_RC=4, one or more data sets may have been excluded from the copy due to a warning. These are not included in these statistics.

User response:

Determine the cause of the errors and resubmit the job.

CKZ54201I **All attempted COPY | Simulate**
requests completed without
errors, table spaces
***num_of_table_spaces_ok*, index**
spaces *num_of_index_spaces_ok*

Explanation:

This message is informational.

User response:

No action is required.

CKZ54202W **COPY aborted, one or more copies**
may not complete

Explanation:

This message is a warning. Copies may have been aborted due to a previous error

User response:

If unable to resolve the error that caused this warning, contact IBM Software Support.

CKZ54203E **All COPY requests completed with**
errors, TSok
***num_of_table_spaces_ok*, ISok**
***num_of_index_spaces_ok*, TSerr**
***num_of_table_spaces_with_errors*,**
ISerr
num_of_index_spaces_with_errors

Explanation:

This message is an error because MAX-COPY-RC=0 or MAX-COPY-RC=4.

User response:

Determine the cause of the errors and resubmit the job.

CKZ54225I **Begin IMAGE COPY completion**
status report

Explanation

This message is informational. It begins a report of data sets processed. Report fields are as follows:

RC

Processing return code; n/a (not attempted) or 1 (copy processing began, but no return code was detected. Either the actual copy was not attempted, or no copy return code was found for this data set).

SPACE TYPE

TS (table space), IS (indexspace), LS (LOB space), or XS (XML space).

CLONE

blank (if not cloned), Y (cloned space) or B (base space).

REPL DSN

Replace target data set with this copy; Y (target exists) or N (target does not exist).

TRG OBJ

Target object exists; Y (target exists) or N (target does not exist).

OBJ XLATE

DB (database name translation result), TS (table space name translation result), IS (indexspace name translation result), IXC (index creator translation result), IXN (index name translation result), blank (no translation), - (translation match, but not used), or + (translation matched and used).

User response:

No action is required.

CKZ54226I End IMAGE COPY completion status report
Explanation:

This message is informational and ends the report of data sets processed.

User response:

No action is required.

CKZ54227I Begin VSAM to VSAM COPY completion status report
Explanation

This message is informational. It begins a report of data sets processed. Report fields are as follows:

RC

Processing return code; n/a (not attempted) or 1 (copy processing began, but no return code was detected. Either the actual copy was not attempted, or no copy return code was found for this data set).

SPACE TYPE

TS (table space), IS (indexspace), LS (LOB space), or XS (XML space).

CLONE

blank (if not cloned), Y (cloned space) or B (base space).

REPL DSN

Replace target data set with this copy; Y (target exists) or N (target does not exist).

TRG OBJ

Target object exists; Y (target exists) or N (target does not exist).

OBJ XLATE

DB (database name translation result), TS (table space name translation result), IS (indexspace name translation result), IXC (index creator translation result), IXN (index name translation result), blank (no translation), - (translation match, but not used), or + (translation matched and used).

User response:

No action is required.

CKZ54228I End VSAM to VSAM COPY completion status report
Explanation:

This message is informational and ends the report.

User response:

No action is required.

CKZ54300I The following object mismatches were detected
Explanation

This message is followed by one of the following messages, depending on type of mismatch:

- I/W/E. *mismatch_type*, source object *src_qualifier_1.src_qualifier_2*, target object *trg_qualifier_1.trg_qualifier_2*
- I/W/E. *mismatch_type*, source object *src_qualifier_1.src_qualifier_2*, target object *trg_qualifier_1.trg_qualifier_2*, partition *part_num*
- I/W/E. *mismatch_type*, source object *src_qualifier_1.src_qualifier_2*, target object *trg_qualifier_1.trg_qualifier_2*, src column *src_column_name*, trg column *trg_column_name*

In addition, the following message with possible values and a corresponding severity is displayed: source value = *source_value*, target value = *target_value* The severity for the pair of messages depends on the values of parameters OBJECT-MISMATCH-RETURN-CODE and UNLOAD-LOAD-ENABLE. Each of these messages can be followed by one of the following messages:

- Ignored, may be fixed by re-creating, will be checked on re-create compatibility later. The object was marked for re-creation.

- Ignored, may be fixed by adding partitions. Partitions will be added to the table space to resolve the mismatch.
- Ignored, may be fixed after DDL generation. If PROCESS-TYPE(Y) is used, some of the mismatches might be fixed by DDL generation; therefore, those mismatches are ignored during comparison before DDL generation.
- Ignored, may be fixed by UNLOAD/LOAD, will be checked on UNLOAD/LOAD compatibility later. The object was marked for UNLOAD/LOAD.
- Ignored per LONGVAR-COMPATIBILITY(Y). If the source and target columns have different column types, but these types are VARCHAR and LONGVAR, a mismatch is ignored if LONGVAR-COMPATIBILITY is set to Y.
- Ignored due to ALLOW-COPY-ON-MISMATCH(Y). ALLOW-COPY-ON-MISMATCH(Y) is specified. Object processing continues despite the mismatch.
- I. Source Table *creator.name*, Column *colname*, SYSCOLUMNS FOREIGNKEY Override using DATASUBTYPE command, was *old_value*, now *new_value*. The new FOREIGNKEY value will be used to mask data for this column.
- I. Source Table *src_creator.src_name*, Column *src_colname*, Target Table *trg_creator.trg_name*, Column *trg_colname* both SYSCOLUMNS FOREIGNKEY values blank, use Product Default, S (SBCS). Both foreign key values are blank. The product default S (SBCS) will be used to mask data for this column.
- I. Source Table *src_creator.src_name*, Column *src_colname*, Target Table *trg_creator.trg_name*, Column *trg_colname*, Source SYSCOLUMNS FOREIGNKEY value blank, use Target, *trg_value*. The source FOREIGNKEY value is blank. The target FOREIGNKEY value will be used to mask data for this column.
- I. Source Table *src_creator.src_name*, Column *src_colname*, Target Table *trg_creator.trg_name*, Column *trg_colname*, Target SYSCOLUMNS FOREIGNKEY value blank, use Source, *src_value*. The target FOREIGNKEY value is blank. The source FOREIGNKEY value will be used to mask data for this column.
- W. Source Table *src_creator.src_name*, Column *src_colname*, SYSCOLUMNS FOREIGNKEY *src_value*, Target Table *trg_creator.trg_name*, Column

trg_colname, SYSCOLUMNS FOREIGNKEY *trg_value*, Not Compatible. The source and target columns may not be compatible. Verify that the source and target values are compatible. If not, ALTER one or both columns as appropriate.

- I. Row format incompatibility bypassed per the IGNORE-RF-MISMATCH-IF-NO-VAR-COLS(Y) Command, DB=*db_name*, TS=*space_name*. If tables of the table space do not contain any variable length columns, and IGNORE-RF-MISMATCH-IF-NO-VAR-COLS is set to Y, a row format mismatch is ignored.
- W/E. Source table *creator.name* has non-zero version, target table does not exist, DDL processing is enabled and target REPAIR execution is disabled. This will lead to versioning mismatch which cannot be fixed automatically. The severity depends on the OBJECT-MISMATCH-RETURN-CODE value for a TB_VERSION mismatch.

User response:

If possible, correct the mismatches and input parameters values if needed, and resubmit the job.

CKZ54301E Object comparison finished with errors

Explanation:

One or more object mismatches that lead to errors were found during object comparison. Db2 Cloning Tool ends processing with an error.

User response:

If possible, correct the mismatches and input parameter values, if needed, and resubmit the job.

CKZ54302W Object comparison finished with warnings

Explanation:

One or more object mismatches that lead to warnings were found during object comparison.

User response:

If possible, correct the mismatches and input parameter values, if needed, and resubmit the job.

CKZ54303W All data sets for the following table spaces and index spaces will not be copied because of object comparison failure

Explanation

This message is followed by messages in the following format:

- For groups that are excluded due to single object mismatch on table space or table:

space_type db_name.space_name: (Group group_number) Mismatch(es) on TS | TB object_qualifier_1.object_qualifier_2

- For groups that are excluded due to single object mismatch on table space partition:

space_type db_name.space_name: (Group group_number) Mismatch(es) on TP object_qualifier_1.object_qualifier_2.partition_number

- For groups that are excluded due to object mismatches on columns of a single table:

space_type db_name.space_name : (Group group_number) Mismatch(es) on columns from TB table_creator.table_name

- For groups that are excluded due to multiple object mismatches or that have an exclusion reason that cannot fit on a single line:

space_type db_name.space_name : (Group group_number) See CKZ54317W below for details

User response:

Review previous messages for more detailed information about object mismatches. If possible, correct the mismatches and input parameter values, if needed, and resubmit the job.

CKZ54304I Compare objects after DDL generation

Explanation:

Db2 Cloning Tool will compare objects after DDL generation.

User response:

No action is required.

CKZ54305I Review UNLOAD/LOAD compatible object groups

Explanation

Db2 Cloning Tool will check objects that are marked for UNLOAD/LOAD to verify that they are UNLOAD compatible. This message is followed by one of the following messages:

- I. Table space *db_name.space_name* and its dependent objects will be processed by UNLOAD/LOAD. The table space and its related objects are compatible with UNLOAD/LOAD.
- W. Table space *db_name.space_name* cannot be processed by UNLOAD/LOAD, because it does not have its base objects in cloning process. LOB and XML table spaces cannot be processed by UNLOAD/LOAD without their base table spaces.

- W. Table space *db_name.space_name* cannot be processed by UNLOAD/LOAD, because it or its dependent/related table space(s)/part(s)/table(s) has(have) mismatch(es) that cannot be processed by UNLOAD/LOAD. If a table space or its related objects are not compatible with UNLOAD/LOAD, the table space and its related objects will not be processed by UNLOAD/LOAD.
- W. Table space *db_name.space_name* cannot be processed by UNLOAD/LOAD, because it contains table with referential constraints. If a table in table space has referential integrity, the table space and its related objects will not be processed by UNLOAD/LOAD.
- W. Table space *db_name.space_name* cannot be processed by UNLOAD/LOAD, because it contains table *tb_creator.tb_name* that has LOB column with max length 32K or more. If a table in a table space has a LOB column with INLINE LENGTH greater than 32K, the table space and its related objects will not be processed by UNLOAD/LOAD.

User response:

If warning messages were received, correct the mismatches and input parameters values and resubmit the job.

CKZ54306I Check versions for compatible objects

Explanation:

Db2 Cloning Tool will check versions for objects that passed mismatch checking.

User response:

No action is required.

CKZ54307W All data sets for the following table spaces and index spaces will not be copied because of catalog processing error

Explanation

This message is followed by messages in the following format:

- For groups that are excluded due to a single Db2 catalog processing error in a group:

space_type db_name.space_name : (Group group_number) Catalog processing error for TS | IS db_name_in_error.space_name_in_error

- For groups that are excluded due to multiple Db2 catalog processing errors:

space_type db_name.space_name : (Group *group_number*) See CKZ54317W below for details

User response:

Review previous messages for possible reasons for catalog processing errors. If unable to resolve this error, contact IBM Software Support.

CKZ54308W All data sets for the following table spaces and index spaces will not be copied because of target data set processing errors

Explanation

This message is followed by messages in the following format:

- For groups that are excluded due to a single target data set processing error in a group:

space_type db_name.space_name : (Group *group_number*) Processing error for *data_set_name*

- For groups that are excluded due to multiple target data set processing errors:

space_type db_name.space_name : (Group *group_number*) See CKZ54317W below for details

User response:

Review previous messages for possible reasons for target data set processing errors. If unable to resolve this error, contact IBM Software Support.

CKZ54309W All data sets for the following table spaces and index spaces will not be copied because target objects do not exist and parameter COPY-IF-NO-DB2-TARGET-OBJECTS is set to N

Explanation

This message is followed by messages in the following format:

- For groups that are excluded because a single object does not exist on the target Db2:

space_type db_name.space_name : (Group *group_number*) No target objects for *space_type db_name.space_name*

- For groups that are excluded because multiple single objects do not exist on the target Db2:

space_type db_name.space_name : (Group *group_number*) See CKZ54317W below for details

User response:

Change the input parameters (enable DDL generation or allow cloning if no target objects exists) and resubmit the job.

CKZ54310I The following encryption-related mismatches were detected

Explanation

This message is followed by messages in the following format:

I/W/E. *mismatch_description*, source data set *src_dsn*, target data set *trg_dsn*
The precise message contents depends on type of mismatch, plus additional messaging with detailed information and corresponding severity. The severity of messages depends on the values of parameters ENCRYPTION-MISMATCH-RETURN-CODE and UNLOAD-LOAD-ENABLE. These messages might be followed by one of the following messages:

- Ignored, may be fixed by UNLOAD/LOAD, will be checked on UNLOAD/LOAD compatibility later. The object was marked for UNLOAD/LOAD.
- Ignored due to ALLOW-COPY-ON-MISMATCH(Y). ALLOW-COPY-ON-MISMATCH(Y) is specified. Object processing continues despite the mismatch.

mismatch_description can be one of following:

- DATA-MOVER failure. The selected DATA-MOVER PGM cannot copy *src_dsn* to *trg_dsn*. An additional message provides information about the reasons. This is a restriction of the selected DATA-MOVER PGM, and cannot be ignored by specifying ENCRYPTION-MISMATCH-RC(0). Additional messages provide information about the reason for the DATA-MOVER PGM failure. Possible explanations are:

- source data set *src_status*, target data set *trg_status*. The selected data mover does not support copying between encrypted and non-encrypted data sets.
- DATA-MOVER PGM *data-mover* does not support DFSMS encryption. The selected data mover does not support encrypted data sets.

- Possible user copy failure. If cloning involves copying *src_dsn* to *trg_dsn* outside of Db2 Cloning Tool, this copy might fail because one of data sets is encrypted and another is not. Additional messages provide information on encryption statuses for both *src_dsn* and *trg_dsn*.
- Target encryption degradation. Cloning will result in reallocating the encrypted *trg_dsn* as non-encrypted.

- Key label compatibility. Cloning will result in *trg_dsn* being encrypted by a key label that will not be usable on target LPAR. Additional messages provide information about the reason for the unusability of the key label. Possible explanations are:

- key label *kl* does not exist on target LPAR. *trg_dsn* will not be usable on the target LPAR because it uses a key label that is not defined there.
- source and target LPARs have incompatible definitions for key label *kl*. *trg_dsn* will not be usable on the target LPAR because *kl* refers to different encryption keys on the source and target LPARs.
- key label *kl* is not usable on target LPAR: RC=*return_code*, RS=*reason_code*. ICSF had returned the specified return code and reason code during key label compatibility checks. Refer to the *Cryptographic Services Integrated Cryptographic Service Facility Application Programmer's Guide* in IBM Knowledge Center for an explanation of the return code and reason code.

User response:

Review all messages regarding encryption-related mismatches and use the provided information to resolve them. Depending on *mismatch_description* and your environment, possible solutions might be using another DATA-MOVER PGM, specifying UNLOAD-LOAD-ENABLE(Y), ignoring encryption mismatches by specifying ENCRYPTION-MISMATCH-RC(0), or contacting your systems programmer. Refer to detailed information about cloning encrypted objects in this user guide. If unable to resolve mismatches, contact IBM Software Support.

CKZ54311W All data sets for the following table spaces and index spaces will not be copied because of encryption-related compatibility failure

Explanation

This message is followed by messages in the following format:

- For groups that are excluded due to a single encryption mismatch:

space_type db_name.space_name : (Group *group_number*) Encryption mismatch on *data_set_name*

- For groups that are excluded due to multiple encryption mismatches:

space_type db_name.space_name : (Group *group_number*) See CKZ54317W below for details

User response:

Review previous messages for more detailed information about encryption mismatches. Correct the mismatches or input parameter values and resubmit the job.

CKZ54312I Key label for target data set dsname changed from kl1 to kl2 during data set copy

Explanation:

SRCIMCPY or SRCVSCPY processing resulted in reallocating data set *dsname* with key label *kl2*. Before the copy began, *dsname* was allocated using key label *kl1*.

User response:

No response is required.

CKZ54313W The following table spaces and index spaces will not be copied due to UNLOAD/LOAD incompatibility

Explanation

This message is followed by messages in the following format:

- For groups that are excluded because a specified table space or table has a single object mismatch that is not compatible with UNLOAD/LOAD:

space_type db_name.space_name : (Group *group_number*) Mismatch(es) on TS | TB *object_qualifier_1.object_qualifier_2*

- For groups that are excluded because a specified table space partition is not compatible with UNLOAD/LOAD:

space_type db_name.space_name : (Group *group_number*) Mismatch(es) on TP *object_qualifier_1.object_qualifier_2.partition_number*

- For groups that are excluded because columns from a single table have mismatches that are not compatible with UNLOAD/LOAD:

space_type db_name.space_name : (Group *group_number*) Mismatch(es) on columns from TB *table_creator.table_name*

- For groups that are excluded because a specified table is not compatible with UNLOAD/LOAD:

space_type db_name.space_name : (Group *group_number*) Table

table_creator.table_name is not compatible with UNLOAD/LOAD

- For groups that are excluded because they are incompatible with UNLOAD/LOAD for multiple reasons or have an UNLOAD/LOAD incompatibility reason that cannot fit on a single line:

space_type db_name.space_name : (Group *group_number*) See CKZ54317W below for details

User response:

Review previous messages for more detailed information about the reasons for UNLOAD/LOAD incompatibility. Correct the mismatches or input parameter values and resubmit the job.

CKZ54314I Checking objects for re-creation compatibility

Explanation

Db2 Cloning Tool will check objects that are marked for re-creation to verify that they can be re-created. This message may be followed by the following messages:

- DDL statements for source *space_type db_name.space_name* cannot be generated because its data sets are user-managed. Objects with user-managed data sets cannot be processed with RECREATE-OBJECTS.
- DDL statements for source table space *db_name.space_name* cannot be generated because it is simple(deprecated). Re-creating the corresponding target object would require generating DDL for a simple table space, which is not possible because simple table spaces are deprecated. DROP statements for the corresponding target object will not be generated.
- Source index *db_name.space_name* and its paired target object cannot be processed with DDL generator because it is implicitly created and its base object is not included in the cloning. Implicit indexes are created only with their base objects.
- Target index *db_name.space_name* cannot be re-created without its base table space which was not included in the cloning. This warning can occur if an index has a mismatch that can be resolved only by re-creating its base object.
- Source table space *db_name.space_name* and its paired target object cannot be processed with DDL generator because it is a LOB object and its base table space was not included in the cloning.

LOB table spaces cannot be processed with RECREATE-OBJECTS without their base table spaces.

- Table *tb_creator.tb_name* cannot be re-created because it is restricted for DROP and RECREATE-TABLE-RESTRICTED-ON-DROP is turned off. If a table is restricted for DROP, and RECREATE-TABLE-RESTRICTED-ON-DROP is turned off, the DROP statement for this table will fail.
- Source object *db_name.space_name* is paired to itself. This warning can occur when the cloning is done within one subsystem and the object will be cloned onto itself.
- Target *space_type db_name.space_name* cannot be re-created because it is in UT* status. The target object is in UTILITY access status. The DROP statement will fail.

User response:

If possible, correct input parameters values or other issues and resubmit the job.

CKZ54315W The following objects are not eligible for re-creation:

Explanation:

This message is followed by a list of objects that require re-creation but cannot be re-created. Members of the list are displayed in the following format: *space_type db_name.space_name*

User response:

Correct the issues that are listed in message CKZ54314I and resubmit the job.

CKZ54316I The following objects will be re-created:

Explanation:

This message is followed by a list of objects that require re-creation. Members of the list are displayed in the following format: *space_type db_name.space_name*

User response:

No action is required.

**CKZ54317W Group *group_number*:
 *message_text***

Explanation

This message might accompany CKZ54303W, CKZ54307W, CKZ54308W, CKZ54309W, or CKZ54311W when the reason for group exclusion cannot fit on a single line. The *group_number* in this message corresponds to the group number that is displayed for excluded objects in CKZ54303W, CKZ54307W, CKZ54308W, CKZ54309W, or

CKZ54311W. The following are possible values for *message_text*:

- Encryption mismatch on *data_set_name_1*
Encryption mismatch on *data_set_name_2*
...
Encryption mismatch on *data_set_name_n*
- Processing error for *data_set_name_1*
Processing error for *data_set_name_2*
...
Processing error for *data_set_name_n*
- No target objects for TS | IS
db_name_1.space_name_1
No target objects for TS | IS
db_name_2.space_name_2
...
No target objects for TS | IS
db_name_n.space_name_n
- Mismatch(es) on TS | TB *object_qualifier_1_1.object_qualifier_2_1*
Mismatch(es) on TS | TB *object_qualifier_1_2.object_qualifier_2_2*
...
Mismatch(es) on TS | TB
object_qualifier_1_n.object_qualifier_2_n
- Mismatch(es) on TP
db_name_1.space_name_1.part_number_1
Mismatch(es) on TP
db_name_2.space_name_2.part_number_2
...
Mismatch(es) on TP
db_name_n.space_name_n.part_number_n
- Mismatch(es) on columns from TB
table_creator_1.table_name_1
Mismatch(es) on columns from TB
table_creator_2.table_name_2
...
Mismatch(es) on columns from TB
table_creator_n.table_name_n
- Catalog processing error for TS | IS
db_name_1.space_name_1
Catalog processing error for TS | IS
db_name_2.space_name_2
...
Catalog processing error for TS | IS
db_name_n.space_name_n
- Table *table_creator_1.table_name_1* is not compatible with UNLOAD/LOAD
Table *table_creator_2.table_name_2* is not compatible with UNLOAD/LOAD
...
Table *table_creator_n.table_name_n* is not compatible with UNLOAD/LOAD

- Mismatch(es) on TB
table_creator_1.table_name_1
Mismatch(es) on TB
table_creator_2.table_name_2
...
Mismatch(es) on TB
table_creator_n.table_name_n

User response

Action depends on the type of error:

- For encryption mismatches, review previous messages for more detailed information about encryption mismatches. Correct the encryption mismatches or input parameter values and resubmit the job.
- For target data set processing errors, review previous messages for possible reasons for target data set processing errors. If unable to resolve the errors, contact IBM Software Support.
- For non-existent target objects, change the input parameters (enable DDL generation or allow cloning if no target objects exist) and resubmit the job.
- For object mismatches, review previous messages for more detailed information about object mismatches. If possible, correct the mismatches and input parameter values, if needed, and resubmit the job.
- For Db2 catalog processing errors, review previous messages to determine the reasons for the catalog processing errors. If unable to resolve this error, contact IBM Software Support.
- For incompatibility with UNLOAD/LOAD, review previous messages for more detailed information about the UNLOAD/LOAD incompatibility. Correct the mismatches or input parameter values and resubmit the job.

CKZ54400E Fatal execute DDL error

Explanation:

This is an error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ54401I DDL output, PGM parm is not NONE, no copies to be performed

Explanation:

DDL PROCESS-TYPE is A. No connection to the target system is made. Copies will not be made.

User response:

If you want to only create source DDL, no action is required. If further processing is required, such as comparison of source and target objects, or template

processing, rerun the job with source DDL generation disabled.

CKZ54402W *space_type* Data set
source_data_set_name is Not
Cataloged on the Source System,
Cannot be Copied

Explanation:

This message is a warning that this data set is not cataloged on the source z/OS system.

User response:

The object exists in the catalog but the data set may be migrated or have been deleted. If migrated, recall the data set and rerun the job. If deleted, use the Exclude keyword to prevent the object from being considered for copy.

CKZ54403W Table space *db_name.space_name*
was excluded because it does not
contain any tables

Explanation:

This message is a warning. The table space was excluded from cloning because the source object from the LISTDEF does not contain any tables; therefore, the table space does not contain any data.

User response:

You can ignore this message or exclude the object via the LISTDEF from cloning to avoid this message.

CKZ54404E *space_type*
database_name.space_name
extracted by LISTDEF utility for
cloning was not found in source
cache

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool during cache population. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ54405E No *object_type* were found in
cache for *space_type*
database_name.space_name
extracted by LISTDEF utility for
cloning

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool during cache population. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ54406E DB *database_name* was not found
in source cache

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool during cache population. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ54407I OBJECT TRANSLATE
translation_parameter IN USE,
SOURCE=*source_creator*,
TARGET=*target_creator*

Explanation:

This message is informational. *translation_parameter* can be DATABASE, TABLESPACE, TABLE, INDEX, INDEXSPACE, CREATOR, or VCAT.

User response:

No action is required.

CKZ54435I *object_type* OBJECT *object_name*
WAS NOT TRANSLATED

Explanation:

This message is informational. It appears when WARN-IF-OBJECT-NOT-TRANSLATED(N). Possible object types are DB, TS, TB, CR, IX, IS, or VC.

User response:

No action is required.

CKZ54438W tt OBJECT ooo WAS NOT
TRANSLATED

Explanation:

This message is a warning. It appears when WARN-IF-OBJECT-NOT-TRANSLATED(Y). Indexspace (IS) may be specified, however, no message will appear if not found. The IX object type message will be printed in its place. tt = object type [DB, TS, TB, CR, IX] ooo = object name

User response:

Verify that all object translations were done correctly. See CKZ56904I in CKZLOG to determine which masks were used for each translation.

CKZ54454I SIMULATE WITHOUT
ALLOCATIONS COMPLETED

Explanation:

This message is informational.

User response:

No action is required.

CKZ54457I COPY REQUESTS TO BE
COMPLETED BY THE USER,
TABLESPACES ttt, INDEXSPACES
iii

Explanation:

This message is informational. ttt = count of table spaces iii = count of indexspaces

User response:

No action is required.

CKZ54463W **DDL output, PGM parm is not NONE, no copies to be performed.**

Explanation:

The DDL PROCESS-TYPE parameter is set to X. Copies will not be performed.

User response:

Rerun the source job using PROCESS-TYPE=X to execute the DDL. Then rerun the source job a second time to perform the copies.

CKZ54465E **Base | Clone table space *database_name.table_space_name.partition* | extension number was not included in any consistent FlashCopy copies.**

Explanation:

A consistent FlashCopy for the specified table space was not found.

User response:

Create a consistent FlashCopy for the specified table space.

CKZ54466W **Base | Clone table space *database_name.table_space_name.partition* | extension number was not included in last consistent FlashCopy. Copy may result in data inconsistencies in target DB2.**

Explanation:

One or more copied table spaces have newer consistent FlashCopy copies than the specified table space.

User response:

If you are sure that using an older consistent FlashCopy for the specified table space will not cause data inconsistencies, no action is required. Otherwise, ensure that consistent FlashCopy copies for all copied table spaces are created within a single COPY utility invocation. Note that if a base/clone relationship between copied table spaces exists, this message will be issued for one of its instances, depending on the copy creation order for base/clone instances.

CKZ54467I **Base | Clone index space *database_name.indexspace.partition* | extension number was not included in any consistent FlashCopy copies and will be rebuilt.**

Explanation:

No consistent FlashCopy copies were found for the specified index space. The specified index space will be rebuilt in the target job.

User response:

If you do not want to rebuild the specified index, a consistent FlashCopy for the index must be created along with the related table space. Otherwise, no action is required.

CKZ54468I **Base | Clone index space *database_name.index_space_name.partition* | extension number consistent FlashCopy PIT_RBA does not match PIT_RBA of its table space consistent FlashCopy. Index space will be rebuilt.**

Explanation:

PIT_RBA values for the specified index space and the associated consistent FLASHCOPY for the associated table space are different. The index will be rebuilt to keep the data consistent on the target DB2.

User response:

If you do not want to rebuild the specified index, a consistent FlashCopy for the index must be created along with the related table space. Otherwise, no action is required.

CKZ54501E **LENGTH EXCEEDED, STRING sss**

Explanation:

This is a user error. sss = string with bad length

User response:

Correct the input and resubmit the job.

CKZ54502E **FIELD DELIMITER ERROR, STRING sss**

Explanation:

This is a user error. sss = string with the error

User response:

Correct the input and resubmit the job.

CKZ54503E **ILLEGAL CHARACTER IN STRING sss**

Explanation:

This is a user error. sss = string with the error

User response:

Correct the input and resubmit the job.

CKZ54504W **ONE OR MORE DATASETS MAY NEED TO BE SYNCED TO THE TARGET DB2, NO SYNCDB2 DD IS AVAILABLE**

Explanation:

This is a warning.

User response:

Add the SYNCDB2 DD to the source job and resubmit the job.

CKZ54521I **BEGIN SOURCE TABLESPACE DATASET REPORT**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ54522I TABLESPACE OBJECT COUNT: ddd,
 DATASET COUNT: ddd**

Explanation:

This message is informational. ddd = decimal count

User response:

No action is required.

**CKZ54523I BEGIN SOURCE INDEXSPACE
 DATASET REPORT**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ54524I INDEXSPACE OBJECT COUNT:
 ddd, DATASET COUNT: ddd**

Explanation:

This message is informational. ddd = decimal count

User response:

No action is required.

**CKZ54525I END SOURCE TABLESPACE/
 INDEXSPACE DATASET REPORT**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ54527I DSN=*dsname* Type= table space |
 index space Obj=*object_type*
 object_qualifier_1.object_qualifier_
 *2***

Explanation:

This message is informational.

User response:

No action is required.

CKZ54528I DSN=*dsname*

Explanation:

This message is informational.

User response:

No action is required.

**CKZ54529I Table | AuxTb | XmlTb *table_type*
 *table_creator.table_name***

Explanation:

This message is informational. *table_type* is one of the following table types: blank if base table with no clone;

B if a base table with a clone; C if a clone table, M if a materialized query table.

User response:

No action is required.

**CKZ54531I CKZC OUTPUT OPEN OK FOR
 DDNAME *ddname***

Explanation:

This message is informational. CKZC is the COPY parameter DATASETS-TO-COPY-DDN data set.

User response:

No action is required.

**CKZ54532W CKZC OUTPUT OPEN FAILED FOR
 DDNAME *ddname*, RC=*rrr*, RS=*sss*,
 USE CKZPRINT TO GET DATASET
 NAMES**

Explanation:

This message is a warning that the DDName pointed to by the COPY parameter DATASETS-TO-COPY-DDN could not be opened. *rrr* = return code *sss* = reason code

User response:

Correct the CKZC data set and resubmit the job or get the data sets to copy from CKZPRINT.

**CKZ54541I CKZS OUTPUT OPEN OK FOR
 DDNAME *ddname***

Explanation:

This message is informational. CKZS is the COPY parameter SYNCDB2-DDN data set.

User response:

No action is required.

**CKZ54542E CKZS OUTPUT OPEN FAILED FOR
 DDNAME *ddname*,
 RC=*return_code*, RS=*reason_code***

Explanation:

This message is a warning that the DD name pointed to by the COPY parameter SYNCDB2-DDN could not be opened.

User response:

Correct the CKZS data set and resubmit the job.

**CKZ54544I OBJID *ooo*=*hhh* FOR *ttt nnn.sss*
 PART(*ppp*) IS NOT CHANGING, NO
 XLATE COMMAND REQUIRED**

Explanation:

This message is informational. *ooo* = object ID field name *hhh* = hexadecimal object ID value *ttt* = data set type, TS or IS *nnn* = database name *sss* = space name *ppp* = partition number

User response:

No action is required.

CKZ54558I *parameter_value*

Explanation:

This message contains hard-coded log apply parms.

User response:

No action is required.

CKZ54589W **ONE OR MORE OBJECT DATASETS
COPIED WITHOUT A TARGET
OBJECT, NO IDCAMS_DDN WAS
AVAILABLE**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ54595I **OUTPUT OPEN OK FOR DDNAME
ddname**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54596W **OUTPUT OPEN FAILED FOR
DDNAME ddname, RC=rrr, RS=sss**

Explanation:

This is a user error. rrr = return code sss = reason code

User response:

Correct the input and resubmit the job.

CKZ54601I **NO DB2 VSAM DATASET WRITES,
SET SCAN-ONLY (Y) IS IN EFFECT**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54602I **Total page set IO:
reads=reads_in_hex,
writes=writes_in_hex**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54603I **TOTAL PAGE SET IO:
PGRK=page_reads,
PGWK=page_writes,
PGRE=page_reads,
PTWE=page_write_errors**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54604I **NO XML UPDATES PERFORMED,
SET SCAN-ONLY(Y) IS IN EFFECT**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54605I **XMLSTRINGS update finished
successfully on subsystem_ID**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54606E **XMLSTRINGS update finished with
errors on subsystem_ID**

Explanation:

This message might occur during update of the target XMLSTRINGS.

User response:

Check the binds on the target system and the availability of SYSXMLSTRINGS. If unable to resolve this error, collect Db2 Cloning Tool target and source server (if used) job output and contact IBM Software Support.

CKZ54607E **REPAIR error, RC=return_code
RS=reason_code**

Explanation:

The REPAIR utility returned an error.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ54608E **REBUILD error, RC=return_code
RS=reason_code**

Explanation:

The REBUILD utility returned an error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ54609E **Unmatched indexes processing
error, RC=return_code
RS=reason_code**

Explanation:

This is a Db2 error or an internal error.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ54611W **ALL TARGET DATASET CHANGES
COMPLETE, WITH ERRORS, TSOK**

aaa, ISOK bbb, TSERR ccc, ISERR ddd

Explanation:

This message is a warning. aaa = count of table spaces ok bbb = count of indexspaces ok ccc = count of table spaces with errors ddd = count of indexspaces with errors

User response:

Determine the cause of the error(s) and resubmit the job.

CKZ54612I ALL TARGET DATASET sss COMPLETED WITHOUT ERRORS, TABLESPACES aaa, INDEXSPACES bbb

Explanation:

This message is informational. sss = Changes or Scans aaa = count of table spaces ok bbb = count of indexspaces ok

User response:

No action is required.

CKZ54613E Error reading XMLSTRINGS on subsystem_ID

Explanation:

This message can occur while reading the source XMLSTRINGS.

User response:

Check the binds on the source system and the availability of SYSXMLSTRINGS. If unable to resolve this error, collect Db2 Cloning Tool target and source server (if used) job output and contact IBM Software Support.

CKZ54615I number_of_lines XMLSTRINGS lines read on subsystem_ID

Explanation:

This message is informational.

User response:

No action is required.

CKZ54623I CURRENTLY EXISTING XMLSTRING IDS MAPPED: SRC=sss, TRG=ttt, ssss

Explanation:

This message is informational. Source string IDs are passed from the source job in a data set. Target string IDs include those read from the target catalog and newly added string values that do not exist in the target catalog but do exist in the source catalog. sss = hex source string ID ttt = hex target string ID ssss = first 80 bytes of the string value

User response:

No action is required.

CKZ54625I Selected for REPAIR, SPACE-TYPE=space_type, DSN=data_set_name, RowFMT=row_format, Version=version, EXTFMT=extended_format, HASHPAGES=hash_data_pages, ALL=all

Explanation:

This message is informational. row_format is set to Y if the REPAIR is for row format change; version is set to Y if the REPAIR is for version change, extended_format is set to Y if the REPAIR is for extended format change; hash_data_pages is set to Y if the REPAIR is for HASHDATAPAGES change, all is set to Y if the REPAIR is for all changes.

User response:

No action is required.

CKZ54626I REPAIR space_type counts, RowFMT=num_of_row_format_changes, Version=num_of_version_changes, EXTFMT=num_of_extended_format_changes, HASHPAGES=num_of_hash_data_page_chgs, All=num_all_repairs

Explanation:

This message is informational. num_of_row_format_changes is the number of REPAIRS for row format changes; num_of_version_changes is the number of REPAIRS for version changes, num_of_extended_format_changes is the number of REPAIRS for extended format changes; num_of_hash_data_page_chgs is the number of REPAIRS for HASHDATAPAGES changes, num_all_repairs is the number of REPAIRS for all changes.

User response:

No action is required.

CKZ54627I Strings found=number_of_XML nodes_found_in_hex, strings changed=number_of_XML string IDs_changed_in_hex

Explanation:

This message provides information about XML processing.

User response:

No action is required.

CKZ54628I XML pages printed=number_printed

Explanation:

This message provides information about the number of XML pages that were printed.

User response:

No action is required.

CKZ54629I **There are no objects that can be processed by LOG-APPLY, LOG-APPLY-ENABLE set to N**

Explanation:

There are no data sets passed in the target job with SYNCDB2 that can be processed by LOG-APPLY. LOG-APPLY functionality will not be used.

User response:

No action is required.

CKZ54640I **Begin SCANONLY status report**

Explanation

This message precedes a report that provides information about the data sets scanned in the target job. The following columns are provided in the report:

TARGET DATASET

The name of the target data set.

RC

The 2-digit return code from data set processing.

SPACE TYPE

The space type: IS (index space) or TS (table space).

PAGES

The total number of pages in the data set.

LOG PAGES CHANGED

The number of pages that had one or more log records applied.

DATA MASKING PAGES CHANGED

The number of pages that had one or more data masking changes applied.

VSAM READS

The total number of VSAM reads, including zero pages.

VSAM WRITES

The total number of changed pages.

IO ERR

This column contains Y if an I/O error occurred.

VSAM ERR

This column contains Y if a VSAM error occurred

EXT FMT

This column contains Y if the page set has a 10-byte RBA/LRSN, or a blank if the page set has a 6-byte RBA/LRSN.

SECS

The elapsed time to process the data set in seconds.

User response:

No action is required.

CKZ54641I **Begin completion status report.**

Explanation

This message precedes a report that provides information about the data sets processed in the target job. The following columns are provided in the report:

TARGET DATASET

The name of the target data set.

RC

The 2-digit return code from data set processing.

SPACE TYPE

The space type: IS (index space) TS (table space), LS (LOB table space), or XS (XML table space).

PAGES

The total number of pages in the data set.

LOG PAGES CHANGED

The number of pages that had one or more log records applied.

DATA MASKING PAGES CHANGED

The number of pages that had one or more data masking changes applied.

VSAM READS

The total number of VSAM reads, including zero pages.

VSAM WRITES

The total number of changed pages.

IO ERR

This column contains Y if an I/O error occurred.

VSAM ERR

This column contains Y if a VSAM error occurred

EXT FMT

This column contains Y if the page set has a 10-byte RBA/LRSN, or a blank if the page set has a 6-byte RBA/LRSN.

SECS

The elapsed time to process the data set in seconds.

User response:

No action is required.

CKZ54642I **End SCANONLY status report**

Explanation:

This message is informational and ends the report.

User response:

No action is required.

CKZ54643I **End completion status report**

Explanation:

This message is informational and ends the report.

User response:

No action is required.

CKZ54644E Remote connect not completed, error_type

Explanation:

This message is either a user error or a system error. Db2 Cloning Tool is unable to connect to the target subsystem.

User response:

If the error type is UNKNOWN, contact IBM Software Support. If the error type is ERROR, determine the cause of the error(s) and resubmit the job. This could be a parameter error. If not, contact your systems programmer or data base administrator.

CKZ54645I Remote connect completed, using connect_type

Explanation:

This message is informational.

User response:

No action is required.

CKZ54650I Begin source copy completion status report

Explanation

This message precedes a report in the target job that provides information about the data sets processed in the source job. The following columns are provided in the report:

TARGET DATASET

The name of the target data set.

RC

The 2-digit return code from data set processing.

SPACE TYPE

The space type: IS (index space) or TS (table space).

PAGES

The total number of pages in the data set.

LOG PAGES CHANGED

The number of pages that had one or more log records applied.

DATA MASKING PAGES CHANGED

The number of pages that had one or more data masking changes applied.

VSAM READS

The total number of VSAM reads, including zero pages.

VSAM WRITES

The total number of changed pages.

IO ERR

This column contains Y if an I/O error occurred.

VSAM ERR

This column contains Y if a VSAM error occurred

EXT FMT

This column contains Y if the page set has a 10-byte RBA/LRSN, or a blank if the page set has a 6-byte RBA/LRSN.

SECS

The elapsed time to process the data set in seconds.

User response:

No action is required.

CKZ54651I End source copy completion status report

Explanation:

This message is informational and ends the report.

User response:

No action is required.

CKZ54700I Some source indexes were excluded because they require REBUILD. INTELLIGENT-REBUILD DD should be specified in the target job to generate the correct statements for REBUILD, because REBUILD-INDEXES-EXECUTE is disabled

Explanation:

If REBUILD-INDEXES-EXECUTE is disabled, the CKZINTRB DD should be specified in the target job to allow Db2 Cloning Tool to generate the correct statements for target REBUILD.

User response:

If not already included, specify the CKZINTRB DD in the target job JCL. Otherwise, no action is required.

CKZ54701E QUIESCE error, RC=return_code RS=reason_code

Explanation:

The QUIESCE utility returned an error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ54702W TARGET SS Db2_subsystem, OBJECT IN DB2 CATALOG, object_type database_name.space_name, WITH DEFINE NO, the CKZ 5th level qualifier, F0001, will be used for the new target data set

Explanation:

This message is a warning.

User response:

Run the target job to change OBIDs and reset LOGRBAs. Create the target data set by inserting dummy data. Then run IDCAMS using the IDCAMS input statements to delete the dummy data set and rename the copied data set.

CKZ54703I Specified **TARGET-PREFETCH-DATABASE-LIST** will not be used because **TCP connection is not used to work with target catalog. Internal database list will be used for target prefetch**

Explanation:

When a CAF or DDF connection is in use, an internal database list is used to populate the cache only for databases that contain objects that are cloned (with OBJECT-TRANSLATE rules applied, if specified).

User response:

No action is required.

CKZ54704I **TARGET SS ssss, OBJECT NOT IN DB2 CATALOG, tt ddd.sss, PARMS ALLOW COPY**

Explanation:

This message is informational. ssss = DB2 subsystem tt = object type (TS or IS) ddd = database name sss = space name

User response:

No action is required.

CKZ54706E **TABLE *creator.table_name* CANNOT BE MATCHED, OBIDS CANNOT BE TRANSLATED**

Explanation:

This message indicates that the table OBIDs in the source and target table spaces cannot be mapped and thus cannot be changed. The table names must be the same on source and target or must be mapped with OBJXLATE.

User response:

Change the table names to match or use OBJXLATE to map the names and resubmit the job.

CKZ54707I **SOURCE TABLE *ccc.nnn* IN TABLESPACE *ddd.sss* AND TARGET TABLE *ccc.nnn* IN TABLESPACE *ddd.sss* ARE PAIRED FOR OBID TRANSLATION**

Explanation:

This message is informational. ccc = creator nnn = table name ddd = database name sss = space name

User response:

No action is required.

CKZ54709I **TARGET SS ssss, tt OBJECT WITH SOURCE DSN=*dsname* AND TARGET DSN=*dsname*, PARMS SPECIFIED PGM NONE, USER IS RESPONSIBLE FOR THE COPY TO THE TARGET**

Explanation:

This message is informational. ssss = DB2 subsystem tt = object type (TS or IS)

User response:

No action is required.

CKZ54710W **TARGET SS *tttt*, tt SOURCE OBJECT *ddd.ttt*, PARTITION *ppp* SOURCE OBJECT PARTED, TARGET NOT PARTED, WILL NOT BE COPIED**

Explanation:

This message is a warning that no copy will be attempted. tttt = target DB2 subsystem tt = object type (TS or IS) ddd = source database name ttt = source space name ppp = source partition number

User response:

Correct the object(s) and resubmit the job.

CKZ54711E **TEMPLATE in DD *ddname* is invalid**

Explanation:

Errors were encountered during verification of the TEMPLATE from the DD *ddname*. For detailed information, see CKZ71600I or CKZ71616W messages.

User response:

Correct the errors in the TEMPLATE using information from DSNUTILU or DSNUTILV messages and resubmit the job.

CKZ54712I **Index *creator_name* (*database_name.index_space_name*) part *part_num* requires a REBUILD, data set will not be copied to target**

Explanation:

An index partition was excluded from the cloning process because it requires a REBUILD to be run on the target. If REBUILD-INDEXES-EXECUTE is enabled, the index will be rebuilt. Otherwise, CKZINTRB DD should be specified in the target job to allow Db2 Cloning Tool to correctly generate statements for the target REBUILD.

User response:

If not already included, specify CKZINTRB DD in the target job JCL. Otherwise, no action is required.

CKZ54713I **Index *creator_name* (*database_name.index_space_name*)**

e) part *part_num*, instance instance(s), requires a REBUILD, data sets will not be copied to target

Explanation:

Instances of an index partition were excluded from the cloning process because REBUILD is required to be run on the target. If REBUILD-INDEXES-EXECUTE is enabled, the index will be rebuilt. Otherwise, CKZINTRB DD should be specified in the target job to allow Db2 Cloning Tool to correctly generate statements for the target REBUILD.

User response:

If not already included, specify CKZINTRB DD in the target job JCL. Otherwise, no action is required.

CKZ54714W tt DATA SET ddd IS NOT CATALOGED ON THE TARGET SYSTEM, AND THE OBJECT EXISTS, WILL BE COPIED TO THE TARGET SUBSYSTEM USING CKZ 5TH LEVEL QUALIFIER, F0001

Explanation:

This message is a warning that this data set is not cataloged on the target z/OS system. tt = data set type (TS or IS) ddd = source data set name

User response:

Run the target job to change OBIDs and reset log RBAs. Recall migrated data set(s) or use IDCAMS to allocate missing data set(s), then run IDCAMS using the IDCAMS input statements to delete the dummy data set(s) and rename the copied data set(s).

CKZ54715W TARGET SS ssss, tt OBJECT WITH SOURCE DSN=dsname, NO TARGET OBJECTS FOUND AND NO DEFAULT VCAT SPECIFIED, COPY CANNOT BE PERFORMED

Explanation:

This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS)

User response:

Correct the error and resubmit the job.

CKZ54716E Target SS *subsystem_id*, *object_type* object *database_name.space_name*, target VCAT is reserved for DB2 system data sets, cannot copy.

Explanation:

This message is an error.

User response:

Correct the input and resubmit the job.

CKZ54717I *space_type* data set *data_set_name* page size is *page_size* and CISIZE is *ci_size*

Explanation:

This message is informational.

User response:

No action is required.

CKZ54718W tt DATASET dsname PAGE SIZE IS ddd AND CISIZE IS eee

Explanation:

This message is a warning and is output only when FUZZY-COPY is Yes. tt = object type (table space or index space) dsname = data set name ddd = page size in decimal bytes eee = VSAM CISIZE size in decimal bytes

User response:

Submit the source job with FUZZY-COPY(N) or change the CISIZE of the data set to equal the page size.

CKZ54719W tt dd.ss STATUS IS sss, SUBSYSTEM ssss

Explanation:

This message is informational. tt = space type (table space or index space) dd = database ss = space name sss = status value ssss = source or target subsystem name

User response:

Determine the cause of the status condition and correct if necessary.

CKZ54720W tt dd.ss.pp STATUS IS sss, SUBSYSTEM ssss

Explanation:

This message is informational. tt = space type (table space or index space) dd = database ss = space name pp = partition number sss = status value ssss = source or target subsystem name

User response:

Determine the cause of the status condition and correct if necessary.

CKZ54721W All target catalog accesses completed, with errors, *space_type*, ok *number_of_spaces_ok*, err *number_of_spaces_with_errors*

Explanation:

This message is informational. The message lists the space type (table spaces or index spaces), how many spaces were accessed without errors, and how many spaces encountered access errors.

User response:

Determine the cause of the error(s) and resubmit the job.

CKZ54722I **All Target Catalog Accesses Completed Without Errors, space_type number_of_spaces_ok**

Explanation

This message is informational and displays the type and number of spaces that were accessed without error.

Note: A space that is not found is not counted as an error.

User response:
No action is required.

CKZ54723I **tt STATUS VALUE sss IS INVALID FOR DB2 VERSION vvv, DISCARDED**

Explanation:
This message is informational. tt = status type ADVISORY or RESTRICT sss = status value vvv = DB2 version

User response:
No action is required.

CKZ54725W **DATASET dsname, WILL NOT BE COPIED DUE TO SOURCE AND/OR TARGET STATUS FAILURE**

Explanation:
This message is a warning. dsname = data set name

User response:
Correct the status failure and resubmit the job.

CKZ54726E **Fatal generate DDL error**

Explanation:
This is an error.

User response:
Contact IBM Software Support if unable to resolve this error.

CKZ54728I **RESTRICT STATUS added per QUIESCE requirement for log apply, status_added**

Explanation:
If a QUIESCE is requested per log apply parm, this status check has been added.

User response:
No action is required.

CKZ54729W **No QUIESCE to be done for data set data_set, disallowed per status value(s), failing_status**

Explanation:

A QUIESCE is requested per log apply parm and the status check has failed. WARN-IF-SKIP-QUIESCE is YES.

User response:
Resolve the status and resubmit the job.

CKZ54730I **No QUIESCE to be done for data set data_set, disallowed per status value(s), failing_status**

Explanation:
A QUIESCE is requested per log apply parm and the status check has failed. In addition, WARN-IF-SKIP-QUIESCE is NO.

User response:
No action is required.

CKZ54731I **OBJECT tt ON TARGET ssss, FOUND DATASET dsname**

Explanation:
This message is informational. tt = object type (TS or IS) ssss = DB2 subsystem

User response:
No action is required.

CKZ54732I **Object object_type on Target target_subsystem, No Object Available, Instance Defaulting for Dataset target_dsname**

Explanation:
This message is informational. The target data set instance defaults to 1.

User response:
No action is required.

CKZ54733W **Object object_type on Target target_subsystem, Cannot find the Table space for the Index, Instance Defaulting for Data set dataset_name**

Explanation:
This message is informational. The target data set instance defaults to 1.

User response:
No action is required.

CKZ54738I **Simple Tablespace Found, sss, ccc, ooo**

Explanation:
This message is informational. A simple tablespace was found in the catalog. This message prints when COPY subcommand WARN-ON-SIMPLE-TABLESPACE is N. sss = subsystem ccc = creator of the object ooo = name of the object

User response:

When copied the first time, ensure the target object is accessible.

CKZ54739W **Simple Tablespace Found, sss, ccc, ooo**

Explanation:

This message is a warning. A simple table space was found in the catalog. This message prints when COPY subcommand WARN-ON-SIMPLE-TABLESPACE is Y. sss = subsystem ccc = creator of the object vvv = name of the object

User response:

When copied the first time, ensure the target object is accessible.

CKZ54741I **Target SS *target_subsystem*, *space_type* object using target DSN=*data_set_name*, IPREFIX I/J switch performed per parms**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54742I **TARGET SS *ssss*, *tt* OBJECT WITH SOURCE DSN=*dsname*, TARGET DSN=*dsname* CAN BE COPIED AS BOTH IPREFIX I/J EXIST ON THE TARGET AND THE PARMS ALLOW DATASET REPLACEMENT**

Explanation:

This message is informational. *ssss* = DB2 subsystem *tt* = object type (TS or IS)

User response:

No action is required.

CKZ54744I **Target SS *target_subsystem*, *space_type* object with source DSN=*source_data_set_name* and target DSN=*target_data_set_name*, can be copied, no target data sets exist, I to J switch has been performed on the target data set**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54745I **TARGET SS *ssss*, *tt* OBJECT WITH SOURCE DSN=*dsname* AND TARGET DSN=*dsname*, CAN BE COPIED AS THE PARMS ALLOW DATASET REPLACEMENT**

Explanation:

This message is informational. *ssss* = DB2 subsystem *tt* = object type (TS or IS)

User response:

No action is required.

CKZ54747I **Target SS *target_subsystem*, *space_type* object with source DSN=*source_data_set_name* and target DSN=*target_data_set_name*, can be copied, no target data sets exist, J to I switch has been performed on the target data set**

Explanation:

This message is informational.

User response:

No action is required.

CKZ54751I **TARGET SS *ssss*, TT OBJECT WITH SOURCE DSN=*dsname* AND TARGET DSN=*dsname*, CAN BE COPIED, NO TARGET DATASETS EXIST, USING IPREFIX DEFAULT F**

Explanation:

This message is informational. *ssss* = DB2 subsystem *tt* = object type (TS or IS)

User response:

No action is required.

CKZ54752W **FOR OBJECT *TS* / *IS* THE SOURCE AND TARGET DATASET NAMES ARE THE SAME, SRCDS = *source_subsystem* *source_data_set_name* AND TRGDS = *target_subsystem* *target_data_set_name***

Explanation:

This message is a warning that the source and target data set names are identical. No copy will be attempted. This can occur when the copy is on the same subsystem and object translation is used such that the object selected by LISTDEF is the target name on an OBJXLATE command.

User response:

Correct the LISTDEF specification and resubmit the job.

CKZ54759W **Dataset Mismatch, Attribute *data_set_attribute*, Source *dsname* = YES / NO Target *dsname* = YES / NO**

Explanation:

This message is a warning. The data set attribute listed in the message does not match between source and target. There is a high probability that the source data set cannot be copied to the target.

User response:

Redefine one or both data sets and resubmit the job.

CKZ54774E **Unable to extend PBG TS, MAXPARTITIONS=*number_partitions*, # PARTITIONS needed=*number_needed*, target TS=*dbname.table_space_name***

Explanation:

The target table space cannot be extended to the same number of partitions as on source because the number of partitions for the target table space is limited by the MAXPARTITION parameter.

User response:

Alter or re-create the target table space with a larger value of MAXPARTITIONS.

CKZ54775I **Extend PBG successful, target TS=*database_name.table_space_name***

Explanation:

This message is informational.

User response:

No action is required.

CKZ54776E **Extend PBG unsuccessful, target TS=*database_name.table_space_name***

Explanation:

An error occurred during ALTER of the target PBG table space. See the output for the possible cause of the error during ALTER execution.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ54780I **Target SSID *target_subsystem_ID*, object in DB2 catalog, *space_type* dsname. No data sets found; will be copied using target IJ indicator per the COPY-IJ-TO-NONEXISTENT-TARGET command**

Explanation:

This message is informational. The target data sets will be created using the target IJ fifth node indicator.

User response:

No response is required.

CKZ54781I **Target SSID *target_subsystem_ID*, object not in DB2 catalog, *space_type* dsname. No data sets found; will be copied using source IJ indicator per the COPY-IJ-TO-NONEXISTENT-TARGET command**

Explanation:

This message is informational. The target data sets are not in the target catalog, therefore the target data sets will be created using the source IJ fifth node indicator.

User response:

No response is required.

CKZ54785I **Number of rotated partitions *number_of_partitions***

Explanation:

This message displays the number of rotated partitions.

User response:

No action is required.

CKZ54791I **ADVISORY-STATUS-VALUES default skipping AREO* and AREOR due to DDL-ENABLE**

Explanation:

This message is informational. When DDL-ENABLE(Y) is specified, the defaults for ADVISORY-STATUS-VALUES are changed to ignore AREO* or AREOR, because these statuses might be expected for certain table spaces. To enable them, specify the ADVISORY-STATUS-VALUES keyword and include AREO* or AREOR, or both.

User response:

No action is required.

CKZ54792E **REMOTE CONNECT NOT COMPLETED, *ccc***

Explanation:

This message is either a user error or a system error. Unable to connect to the target subsystem. *ccc* = error type

User response:

If the error type is UNKNOWN, contact IBM Software Support. If the error type is ERROR, determine the cause of the error(s) and resubmit the job. This could be a parameter error. If not, contact the systems programmer or data base administrator.

CKZ54793I **REMOTE CONNECT COMPLETED, USING *sss***

Explanation:

This message is informational. It may be followed by a failure at the target if the userid or password are invalid. This could be indicated by SQLCODE = -30082. *sss* = connect type (CAF, DDF or TCP/IP)

User response:

No action is required.

CKZ54798W **Source / Target table *table_creator.table_name* cannot be matched, OBIDs cannot be translated**

Explanation:

This message indicates that the table OBIDs in the source and target table spaces cannot be mapped and thus cannot be changed. The table names must be the same on source and target or must be mapped with OBJXLATE.

User response:

Change the table names to match or use OBJXLATE to map the names and resubmit the job.

CKZ547102I **Source | Target table creator.table_name cannot be matched, OBIDs cannot be translated**

Explanation:

This message indicates that the table OBIDs in the source and target table spaces cannot be mapped and thus cannot be changed. The table names must be the same on the source and the target or must be mapped with OBJXLATE.

User response:

Change the table names to match or use OBJXLATE to map the names and resubmit the job if needed.

CKZ547104W **LOG-APPLY cannot be applied to objects processed by UNLOAD-LOAD, that may cause data inconsistency on target**

Explanation:

LOG-APPLY(Y) was specified, but some of the cloned objects need to be processed through UNLOAD-LOAD. Therefore, logs for these objects will not be applied.

User response:

Turn off LOG-APPLY or resolve the mismatch that caused UNLOAD-LOAD processing to allow LOG-APPLY processing and resubmit the job.

CKZ547106W **ALTER statements are not executed because of the simulation mode. This may lead to target data set processing problems during simulation, which are expected to be resolved during non-simulation run**

Explanation:

Problems with target data set processing might occur because simulation mode is in use and there are target PBG table spaces that have less partitions than their paired source table spaces. These problems will not occur during a non-simulation run.

User response:

No action is required.

CKZ547107W **ALTER statements are required for objects with identity columns, but PROCESS-DDL-DDN for DDL**

generation was not specified. This will lead to an error during a non-simulation run

Explanation:

PROCESS-DDL-DDN is required for identity columns ALTER statement processing.

User response:

Specify the required parameter and resubmit the job.

CKZ547108I **Source | Target data set dsname is encrypted with key label "DFSMS_key_label"**

Explanation:

This informational message shows the DFSMS key label of the data set.

User response:

No action is required.

CKZ547109I **Source | Target data set dsname is not encrypted**

Explanation:

This informational message indicates that the data set is not encrypted.

User response:

No action is required.

CKZ547110I **DFSMS encryption is not supported on source | target LPAR**

Explanation:

This informational message indicates that DFSMS on the indicated LPAR does not support DFSMS encryption.

User response:

No action is required.

CKZ547111I **Target data set dsname has different key labels on source LPAR (kls) and target LPAR (klt)**

Explanation:

The source job and the target server job have reported different key labels for *dsname*. This might happen when cloning is performed using an interim set of volumes.

User response:

No action is required.

CKZ547112I **Target data set dsname is accessible from location**

Explanation

location can be one of the following:

- Source LPAR: Only the source job had located *dsname*

- Target LPAR: Only the target server job had located *dsname*
- Source and target LPARs: Both the source job and the target server jobs located *dsname*

User response:

No action is required.

CKZ547114I The RESET-LOGRBA option was automatically set to Y | N

Explanation:

RESET-LOGRBA(A) was specified in the source job and automatically changed to Y or N.

User response:

No action is required.

CKZ54801I FUZZYCPY(Y) SPECIFIED, NO SOURCE SPACES WILL BE STOPPED. USER MUST VALIDATE TARGET DATASET(S)

Explanation:

This message is informational. It is output when COPY parameter FUZZY-COPY is Y.

User response:

No action is required.

CKZ54802W ALL COPY REQUESTS COMPLETED WITH ERRORS, TSOK aaa, ISOK bbb, TSERR ccc, ISERR ddd

Explanation:

This message is informational. aaa = count of table spaces ok bbb = count of indexspaces ok ccc = count of table spaces with errors ddd = count of indexspaces with errors. If MAX_RC=4, one of more data sets may have been excluded from the copy due to a warning. These are not included in these statistics.

User response:

Determine the cause of the error(s) and resubmit the job.

CKZ54803I ALL ATTEMPTED COPY | SIMULATE REQUESTS COMPLETED WITHOUT ERRORS, TABLE SPACES num_of_table_spaces_ok, INDEX SPACES num_of_index_spaces_ok

Explanation:

This message is informational.

User response:

No action is required.

CKZ54804W COPY aborted, one or more copies may not complete.

Explanation:

This message is a warning. Copies may have been aborted due to a previous error.

User response:

Review other error messages. If unable to resolve errors, contact IBM Software Support.

CKZ54805E All COPY requests completed with errors, TSok num_of_table_spaces_ok, ISok num_of_index_spaces_ok, TSerr num_of_table_spaces_with_errors, ISerr num_of_index_spaces_with_errors

Explanation:

This message is an error because MAX-COPY-RC=0 or MAX-COPY-RC=4.

User response:

Determine the cause of the errors and resubmit the job.

CKZ54821I - BEGIN SIMULATE(A) STATUS REPORT

Explanation:

This message is informational. It begins a report of data sets processed. Report fields follow... RC = processing return code SPACE TYPE = TS table space IS indexspace LS LOB space CLONE = blank if not cloned Y is a cloned space (V9 and higher) B is a base space (V9 and higher) REPL DSN = replace target data set with this copy Y target does exist N target does not exist TRG OBJ = target object exists Y target does exist N target does not exist OBJ XLATE DB database name translation result TS table space name translation result IS indexspace name translation result IXC index creator translation result IXN index name translation result blank = no translation - = translation match but not used + = translation match and used

User response:

No action is required.

CKZ54822I - END SIMULATE(A) STATUS REPORT

Explanation:

This message is informational. It ends the report.

User response:

No action is required.

CKZ54823I - BEGIN COPY PGM(NONE) STATUS REPORT

Explanation

This message is informational. It begins a report of data sets processed. Report fields follow:

- RC = processing return code
- SPACE TYPE = TS (table space); IS (indexspace); LS (LOB space)

- CLONE = blank if not cloned; Y is a cloned space (V9 and higher); B is a base space (V9 and higher)
- REPL DSN = replace target data set with this copy; Y = target does exist, N = target does not exist
- TRG OBJ = target object exists; Y = target does exist; N = target does not exist
- OBJ XLATE = DB (database name translation result); TS (table space name translation result); IS (indexspace name translation result); IXC (index creator translation result); IXN (index name translation result) blank = no translation; - = translation match but not used; += translation match and used

User response:
No action is required.

CKZ54824I - END COPY PGM(NONE) STATUS REPORT

Explanation:
This message is informational. It ends the report.

User response:
No action is required.

CKZ54825I - BEGIN COPY COMPLETION STATUS REPORT

Explanation:
This message is informational. It begins a report of data sets processed. Report fields follow... RC = processing return code n/a = not attempted or unable to find a copy return code SPACE TYPE = TS table space IS indexspace LS LOB space CLONE = blank if not cloned Y is a cloned space (V9 and higher) B is a base space (V9 and higher) REPL DSN = replace target data set with this copy Y target does exist N target does not exist TRG OBJ = target object exists Y target does exist N target does not exist OBJ XLATE DB database name translation result TS table space name translation result IS indexspace name translation result IXC index creator translation result IXN index name translation result blank = no translation - = translation match but not used + = translation match and used

User response:
No action is required.

CKZ54826I - END COPY COMPLETION STATUS REPORT

Explanation:
This message is informational. It ends the report.

User response:
No action is required.

CKZ54904E Log apply, unable to match minilog data set with log apply control blocks

Explanation:
This is probably an internal error.

User response:
Contact IBM Software Support.

CKZ54905E Log apply, unable to match space database.table_space_or_indexspace.partition Control Blocks

Explanation:
This is probably an internal error.

User response:
Contact IBM Software Support.

CKZ54907I Number of SPACES-PER-MINILOG changed, now new_number_of_spaces_per_minilog

Explanation:
This message is informational. Only 36 minilog data sets can be allocated. This caused the number of SPACES-PER-MINILOG to be changed from that requested in the LOG-APPLY parameters.

User response:
No action is required.

CKZ54908I RBA/LRSN passed to log apply, rba_lrsn_value

Explanation:
This message is informational.

User response:
No action is required.

CKZ54909E Log apply init call failed, RC=return_code, RS=reason_code

Explanation:
A severe error occurred when log apply was processing a source server request.

User response:
If unable to resolve this error, contact IBM Software Support.

CKZ55002I data_set_type, total DSNs to process = number_of_data_sets, DSNs per utility call = number_of_data_sets_processed_per_call

Explanation:
The QUIESCE utility is split into multiple calls per the UTILITY-COMMAND-EXECUTE-PERCENT SET command. *data_set_type* can be BaseTbl (base table) or CloneTbl (clone table).

User response:
No action is required.

CKZ55003W **No QUIESCE to be done for data set *data_set_name*, disallowed per copy error**

Explanation:

This message is an error.

User response:

Correct the copy error and resubmit the job.

CKZ55100I **VSAM Repository data set open | close, DD *repository_ddname***

Explanation:

The repository DD that is listed in the message was opened or closed.

User response:

No action is required.

CKZ55101E **VSAM Repository data set open | close | read | point | readupd | delete | write | rewrite error, DD *repository_ddname*, LOC=*location***

Explanation:

An error occurred during repository DD processing.

User response:

Contact IBM Software Support. Provide the job log that contains the error and the repository data sets.

CKZ55102E **Wrong repository processing function: *function_name***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support. Provide the job log that contains the error.

CKZ55103E ***Record_type* record not found, LOC=*location***

Explanation:

A required record was not found in the repository data set.

User response:

Contact IBM Software Support. Provide the job log that contains the error and the repository data sets.

CKZ55104E **Runtime repository error, *changed_value(s)* is/are changed**

Explanation:

The values in the repository data set and SYNCDB2 are not the same. SYNCDB2 values cannot be changed for most parameters when the runtime repository is in use.

User response:

If SYNCDB2 values were changed, correct the values and resubmit the job. If unable to resolve this error,

contact IBM Software Support. Provide the job log that contains the error and the repository data sets.

CKZ55105E **Runtime repository write error, DD *repository_ddname*, duplicate record, LOC=*location***

Explanation:

A duplicate record was found when attempting to write to the repository DD.

User response:

Contact IBM Software Support. Provide the job log that contains the error and the repository data sets.

CKZ55106I **Runtime repository, first target job run for specified source job**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55107I **Runtime repository, target job rerun for specified source job**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55108I **Runtime repository, new data was added in *repository_ddname* DD**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55109I **Runtime repository, data was updated in *repository_ddname* DD**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55110I **Runtime repository, compare SYNCDB2 data with *repository_ddname* DD records**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55111I **Runtime repository, compare extents data with CKZRRDSN DD records**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55112I Runtime repository, compare unmatched indexes data with CKZRRJOB DD records

Explanation:

This message is informational.

User response:

No action is required.

CKZ55113I Runtime repository, target log apply, DD *ddname* for BSDS *bsds_data_set* was successfully allocated | freed

Explanation:

This message is informational.

User response:

No action is required.

CKZ55114E Runtime repository, target log apply, DD *ddname* for BSDS *bsds_data_set* allocation | free failed

Explanation:

An error occurred during BSDS data set allocation or free.

User response:

Contact IBM Software Support. Provide the job log that contains the error.

CKZ55115I Runtime repository, target log apply, DD *ddname* for BSDS *bsds_data_set* allocation | free failed, try *second_bsds_data_set*

Explanation:

This message is informational.

User response:

If *second_bsds_data_set* allocation successfully finished, no action is required. If *second_bsds_data_set* allocation failed, contact IBM Software Support. Provide the job log that contains the error.

CKZ55116I BSDS data set open | close, DD *bsds_data_set*

Explanation:

The DD for the BSDS data set that is listed in the message was opened or closed.

User response:

No action is required.

CKZ55117E BSDS data set open | close | read error, DD *bsds_ddname*, LOC=*location*

Explanation:

An error occurred during BSDS data set DD processing.

User response:

Contact IBM Software Support. Provide the job log that contains the error.

CKZ55118E Runtime repository error, RRJP record was not updated in accordance with RRJL record

Explanation:

This is an internal error.

User response:

Contact IBM Software Support. Provide the job log that contains the error and job logs for previous runs with the specified runtime repository data sets for the same source job.

CKZ55200I VSAM repository data set open | close, DD *repository_ddname*

Explanation:

The repository DD was opened or closed.

User response:

No action is required.

CKZ55201E VSAM repository data set open | close error, DD *repository_ddname*, LOC=*location*

Explanation:

An error occurred during repository DD processing.

User response:

Contact IBM Software Support. Provide the job log that contains the error and the repository data sets.

CKZ55202I Report data set open | close, DD *repost_ddname*

Explanation:

The report DD was opened or closed.

User response:

No action is required.

CKZ55203E Report data set open | close error, DD *repost_ddname*, LOC=*location*

Explanation:

An error occurred during report DD processing.

User response:

Contact IBM Software Support. Provide the job log that contains the error and the repository and report data sets.

CKZ55204W Repository DD *repository_ddname* has no records

Explanation:

The data set or job report was requested, but the corresponding repository data set has no records.

User response:

No action is required.

CKZ55205E **Error accessing repository DD
repository_ddname, LOC=location**

Explanation:

An error occurred when attempting to work with the repository data set.

User response:

Contact IBM Software Support. Provide the job log that contains the error and the repository data sets.

CKZ55400I **Number of SPACES-PER-MINILOG
changed, now
recomputed_num_of_spaces_per_
minilog**

Explanation:

This message is informational. Only 36 minilog data sets can be allocated. This caused the number of SPACES-PER-MINILOG to be changed from the number that was specified in the LOG-APPLY parameters.

User response:

No action is required.

CKZ55401W **Source table_space_type table
space database.table_space is not
creating log records**

Explanation:

The table space listed in the message is defined LOGGED NO, so there are no log records to apply. This message only applies when PGM(SRCIMCPY) and is a warning.

User response:

Remove the table space from the LISTDEF, change the definition of the table space to enable logging, or change WARN-IF-TS-DEFINED-LOG-NO(Y) to (N) if logging is not required for this table space. Rerun the job.

CKZ55402I **Source table_space_type table
space database.table_space is not
creating log records**

Explanation:

The table space listed in the message is defined LOGGED NO, so there are no log records to apply. This message only applies when PGM(SRCIMCPY) and is informational.

User response:

No action is required.

CKZ55403I **RBA/LRSN passed to Log Apply,
rba_lrsn_value_in_hex**

Explanation:

This message is informational.

User response:

No action is required.

CKZ55502I **Begin REBUILD INDEXES report**

Explanation

This message is informational. It begins a report of indexes processed by the REBUILD utility. The following columns are provided in the report:

DBNAME

The database name.

SNAME

The table space or indexspace name.

RC

The two-digit return code from utility processing.

SPACE TYPE

TS (table space) or IS (indexspace).

CLONE

B (base instance) or C (clone instance).

PARTED

+ (partitioned object) or blank (non-partitioned object).

PART

n (number of processed partitions), * (all partitions of the object), or blank (non-partitioned object).

UNMATCHED

+ (unmatched index) or blank (not an unmatched index).

User response:

No action is required.

CKZ55503I **End REBUILD INDEXES report**

Explanation:

This message is informational and ends the report of indexes processed by the REBUILD utility.

User response:

No action is required.

CKZ55505I **Begin source excluded indexes
report**

Explanation

A report of indexes excluded from the source job follows this message. The following columns are provided in the report:

- DBNAME: The database name for the index space.
- ISNAME: The index space name.
- TSNAME: The table space name for this index space.
- EXCLUDE REASON: The reason that the index was excluded from cloning. Possible reasons are: "IX_PGFSIZE", "IX_PADDED", "IX VERSION MISMATCH", "NOT IN FCIC", "NOT IN FCIC WITH

TS", "IS EXTENT MISMATCH", "DATA MASKING," or "COPY NO."

- BASE/CLONE: Specifies whether a base or clone instance was excluded. Possible values are: "B" for base instance excluded, "C" for clone instance excluded, or "BOTH" for both clone and base instances excluded.
- PART: The partition number, if a specific partition is excluded; otherwise blank.

User response:

No action is required.

CKZ55506I **The following indexes were not copied and will be rebuilt on target system**

Explanation:

A report of indexes that are excluded from the source job follows this message.

User response:

No action is required.

CKZ55507I **End source exclude indexes report**

Explanation:

This message signifies the end of source exclude indexes report.

User response:

No action is required.

CKZ56000E **DDNAME *ddname* is already allocated**

Explanation:

A DD with the specified name *ddname* is already allocated.

User response:

Correct the input and resubmit the job. If unable to resolve the error, contact IBM Software Support.

CKZ56001E **BPXWDYN error, RC=*return_code*, string=*string***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ56002E **DDNAME(*ddname*), RC=*return_code*, RS=*reason_code*, Open failed**

Explanation:

Db2 Cloning Tool failed to open the specified DD *ddname*.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ56201I **TRACING BUFFER WRAP, POSSIBLE LOOP**

Explanation:

This is an informational WTO. This will only occur if support has requested that tracing be turned on to investigate a problem.

User response:

Increase the size of the tracing buffer and resubmit the job.

CKZ56202E **SUBTASK, CREATION FAILURE, RC=*rrr***

Explanation:

This is probably an internal error. *rrr* = return code

User response:

Contact IBM Software Support.

CKZ56203E **SUBTASK *subtask_number*, IGGCSI00 ERROR RETURN, R15=*return_code*, CSI MODULE ID=*SMS_moduleID*, CSI RETURN CODE IS *SMS_return_code*, CSI REASON CODE IS *SMS_reason_code*, DSN=*requested_data_set_name***

Explanation:

This is probably a catalog error. A subtask number may not be included in the message (or might be 0) if the task that produced this error is a main task.

User response:

See the DFSMS Managing Catalogs publication for your version of z/OS for more information on IGGCSI00. Contact IBM Software Support if unable to resolve this error.

CKZ56204E **Subtask *subtask_number*, IGGCSI00 return work area error, error_type, hex_value_1, hex_value_2, hex_value_3, DSN=*dsname*, Info=*additional_information_about_this_error***

Explanation:

This is probably an internal error.

User response:

See the DFSMS Managing Catalogs publication for your version of z/OS for more information on IGGCSI00. Contact IBM Software Support.

CKZ56205E **Subtask *subtask_number*, Unable to Load Module: *module_name***

Explanation:

This message is an error.

User response:

Ensure the module is available to Db2 Cloning Tool using STEPLIB, LPA, link library, etc.

CKZ56297E Dump for bad IGGCSI00 return work area

Explanation:

A dump for IGGCSI00 return work area follows.

User response:

Contact IBM Software Support.

CKZ56299E Error generating SVC dump

Explanation:

An error occurred while generating a dump for a bad IGGCSI00 return.

User response:

Contact IBM Software Support.

CKZ56401E SUBTASK nn, COMPLETED WITH ERRORS, RC=rrr, RS=sss

Explanation:

This message is an error message. It indicates the subtask has output one or more error messages. Messages output by a subtask will have 'Subtask' followed by a number to identify the issuing subtask. nn = subtask number rrr = return code sss = reason code

User response:

Check CKZPRINT for error messages output by the subtask.

CKZ56402W SUBTASK nn, COMPLETED WITH WARNINGS, RC=rrr, RS=sss

Explanation:

This message is a warning message. It indicates the subtask has output one or more warning messages. Messages output by a subtask will have 'Subtask' followed by a number to identify the issuing subtask. nn = subtask number rrr = return code sss = reason code

User response:

Check CKZPRINT for warning messages output by the subtask.

CKZ56403W SUBTASK nn, COMPLETED WITH WARNING(S)

Explanation:

This message is a warning message. It indicates the subtask has output one or more warning messages. Messages output by a subtask will have 'Subtask' followed by a number to identify the issuing subtask. nn = subtask number

User response:

Check CKZPRINT for warning messages output by the subtask.

CKZ56404I SUBTASK nn, COMPLETED WITH NO ERRORS

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ56497E ABEND 900,REASON=2

Explanation:

This terminates the job with an ABEND.

User response:

Contact IBM Software Support.

CKZ56498E ABEND 900,REASON=1

Explanation:

This terminates the job with an ABEND.

User response:

Contact IBM Software Support.

CKZ56499I TRACE PRINT TERMINATED, WRAPPED

Explanation:

This is an informational WTO. This will only occur if support has requested that tracing be turned on to investigate a problem.

User response:

Increase the size of the tracing buffer and resubmit the job.

CKZ56501E INVALID ERROR ID, MESSAGE DELETED

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ56597E WTO BUFFER ALLOCATION ERROR

Explanation:

This is an error WTO. There is not enough virtual storage.

User response:

Increase the region size and resubmit the job.

CKZ56598E QUEUEING ERROR IN QMSG

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ56599E GETMAIN ERROR IN QMSG

Explanation:

This is an error WTO. There is not enough virtual storage.

User response:

Increase the region size and resubmit the job.

**CKZ56901E INFRA LOAD ERROR, RC=rrr,
RS=sss, MODULE=mmm**

Explanation:

This message is an error message. It indicates support processing cannot load a required module. rrr = return code sss = reason code mmm = module name

User response:

Check the JCL to ensure the STEPLIB is pointing to the correct library and that the install was done correctly. Resubmit the job when corrections made.

**CKZ56902E MASK CHARACTER ERROR, SRC
MASK = sss, TRG MASK = ttt**

Explanation:

This message is an error message. It indicates that the given source target object translation mask pair is invalid. Mask pairs may be invalid for the following reason: %, _, or %_ are in the source and target masks without an interspersed non-mask character. For example: %%% %a%b% is valid %a%b% %%% is valid __. __ is valid .%.%. is an error %_ ._% is an error %_ .%_ is an error sss = source mask value ttt = target mask value

User response:

Correct the mask and resubmit the job.

CKZ56903E Mask error, reason

Explanation:

A masking error occurred during processing of OBJECT-TRANSLATE or DDL-ATTRIBUTE-CHANGE. The reason for the masking error is provided in the message text.

User response:

Correct the input and resubmit the job. If unable to resolve the error, contact IBM Software Support.

**CKZ56904I OBJECT TRANSLATE tt IN USE,
SRC=sss, TRG=ttt, SRC
MASK=mmm , TRG MASK=nnn**

Explanation:

This message is informational. tt = 2 digit object type being translated sss = source object name ttt = target object name mmm = source mask value nnn = target mask value

User response:

No action is required.

**CKZ57301E SUBTASK nn, DETACH ERROR,
CODE=sss**

Explanation:

This is an error message that an MVS task cannot be detached. The return from the detach is in the reason code. nn = subtask number sss = reason code

User response:

Contact IBM Software Support.

CKZ57302I ALL SUBTASKS NOW DETACHED

Explanation:

This message is informational.

User response:

No action is required.

**CKZ57303I SUBTASK nn, DETACH
COMPLETED**

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ57307E NO TASK CONTROL BLOCK

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ57308E NO JOB CONTROL BLOCK

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ57309E NO SAVE AREA

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ57310E NO VIRTUAL STORAGE

Explanation:

This is an error WTO.

User response:

Increase the region size and resubmit the job.

CKZ57501I Subtask *subtask_number*, text

Explanation:

This informational message contains text output of an IDCAMS command

User response:

No action is required.

**CKZ57502I Subtask *subtask_number*, IDCAMS
Messages Follow...**

Explanation:

This message is informational.

User response:

No action is required.

CKZ57503E Subtask *subtask_number*, IDCAMS execute rror, RC=*return_code*, RS=*IDCAMS_reason_code*, text to follow...

Explanation:

This message is an internal error. Additional text may follow this message.

User response:

Contact IBM Software Support.

CKZ57504W Getting attributes for *data_set_name* failed with RC=*return_code*. For more information see server job output

Explanation:

When getting attributes for the specified data set, the Db2 Cloning Tool TCP/IP target server job received a non-zero return code from IDCAMS. Processing continues if MAX_RC is not 0.

User response:

Review CKZ57502I messages in the TCP/IP target server job CKZLOG DD to determine the reason of the IDCAMS failure. If unable to determine or resolve this error, contact IBM Software Support.

CKZ57505E Getting attributes for *data_set_name* failed with RC=*return_code*. For more information, see server job output

Explanation:

When getting attributes for the specified data set, the Db2 Cloning Tool TCP/IP target server job received a non-zero return code from IDCAMS. The cloning process was terminated.

User response:

Review CKZ57502I messages in the TCP/IP target server job CKZLOG DD to determine the reason of the IDCAMS failure. Take appropriate corrective actions and resubmit the source job. If unable to determine or resolve this error, contact IBM Software Support.

CKZ57506W Getting attributes for *data_set_name* failed with RC=*return_code*

Explanation:

When getting attributes for the specified data set, the Db2 Cloning Tool source job received a non-zero return code from IDCAMS. Processing continues if MAX_RC is not 0.

User response:

Review CKZ57502I messages in the source job CKZLOG DD to determine the reason of the IDCAMS failure. If unable to determine or resolve this error, contact IBM Software Support.

CKZ57507E Getting attributes for *data_set_name* failed with RC=*return_code*

Explanation:

When getting attributes for the specified data set, the Db2 Cloning Tool source job received a non-zero return code from IDCAMS. The cloning process was terminated.

User response:

Review CKZ57502I messages in the source job CKZLOG DD to determine the reason of the IDCAMS failure. Take appropriate corrective actions and resubmit the source job. If unable to determine or resolve this error, contact IBM Software Support.

CKZ57508I Subtask *subtask_number*, The data set *data_set_name* was altered.
Candidate_volumes_number_added candidate volumes were added.
Candidate_volumes_number_removed candidate volumes were removed.

Explanation:

This message is informational. The *candidate_volumes_number_added* candidate volumes were successfully added to the *data_set_name* data set. The *candidate_volumes_number_removed* candidate volumes were successfully removed from the *data_set_name* data set.

User response:

No action is required.

CKZ60001I SUBTASK nn, INITIALIZATION COMPLETED

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ60002E SUBTASK nn, ESTAE INIT ERROR, RC=rrr

Explanation:

This is an internal error. nn = subtask number rrr = return code

User response:

Contact IBM Software Support.

CKZ60003I SUBTASK nn, NUMBER OF DATASETS PER COMMAND INCREASED TO ddd

Explanation:

This message is informational. It occurs because the number of data sets per DSS command is low and must be increased due to data set extensions or a

clone table. nn = subtask number ddd = decimal number of the new value

User response:

Consider raising the PARMLIB parameter DSNS_PER_COPY to the maximum of 255.

**CKZ60004E SUBTASK nn, DEQUEING
 ERROR,RC=rrr**

Explanation:

This is an internal error. nn = subtask number rrr = return code

User response:

Contact IBM Software Support.

CKZ60005E ERROR TAKING SVC DUMP: rrr

Explanation:

This is an error WTO. rrr = SDUMP return code

User response:

Contact IBM Software Support.

**CKZ60006I SUBTASK nn, DATASET ddd HAS
 BEEN REQUEUED FOR EXTENSION
 OVERFLOW**

Explanation:

This message is informational. It occurs because a primary DSN and all its extensions or clone table will not fit into the user specified data sets that may be sent to DSS during this call. nn = subtask number ddd = source data set name

User response:

Consider using the default values for DSNS_PER_COPY and DSS_COPY_COMMANDS. This will reduce the occurrences of requeueing.

**CKZ60007I Subtask *subtask_number*, purge
 request, ID=*request_type*,
 XEH=*pointer_to_request_buffer***

Explanation:

This message is informational. It indicates which subtask requests have been purged due to an error.

User response:

No action is required.

CKZ60009E ABEND 400,REASON=9

Explanation:

This is an error WTO. It precedes an abend.

User response:

Contact IBM Software Support.

CKZ60010E ABEND 400,REASON=10

Explanation:

This is an error WTO. It precedes an abend.

User response:

Contact IBM Software Support.

**CKZ60011E ESTAE NO TASK DESCRIPTOR
 ELEMENT**

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ60012E ESTAE NO CONTROL BLOCK

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ60013E ESTAE NO SAVE AREA

Explanation:

This is an error WTO.

User response:

Contact IBM Software Support.

CKZ60014E ERROR TAKING SVC DUMP: rrr

Explanation:

This is an error WTO. rrr = SDUMP return code

User response:

Contact IBM Software Support.

CKZ60015E ERROR TAKING SVC DUMP: rrr

Explanation:

This is an error WTO. rrr = SDUMP return code

User response:

Contact IBM Software Support.

**CKZ61102E SUBTASK nn, DB2 CAF INIT
 ERROR, UNABLE TO LOAD MODULE
 mmm**

Explanation:

This is a probable user or DB2 error. nn = subtask number mmm = name of DB2 load module

User response:

Ensure SDSNLOAD is in steplib. If it is, contact your systems programmer.

**CKZ61104E SUBTASK nn, CANNOT CONNECT
 TO DB2 tttt, PARM FOR sss IS NOT
 AVAILABLE**

Explanation:

This is a probable user or DB2 error. nn = subtask number tttt= target subsystem sss = parm that is not available to TCPIP

User response:

Ensure that CAF or TCPIP can be used for the remote connect. If all the input parms are correct, contact your systems programmer.

CKZ61106E **SUBTASK nn, TCPIP CONNECTION TO THE WRONG SUBSYSTEM, EXPECTING eeee, CONNECTED TO ssss**

Explanation:

This is a user error. The TCPIP server at the IP in the COPY command is not connected to the correct target subsystem for this source job. nn = subtask number eeee= target subsystem expecting from the TCPIP server ssss= target subsystem received from the TCPIP server

User response:

Change the IP on the copy command to a TCPIP server connected to the target subsystem or change the LSSID of the current TCPIP server. Resubmit the server job if required.

CKZ61108I **SUBTASK NN, LOCAL DB2 SUBSYSTEM IN COMPATIBILITY MODE PER DB2**

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ61201E **SUBTASK nn, DB2 SUBSYSTEM ssss NOT ACTIVE, CAFRC=hhh**

Explanation:

The CAF return indicates the DB2 subsystem is not active. nn = subtask number ssss = DB2 subsystem hhh = DB2 call attach facility return code

User response:

Activate the subsystem and resubmit the job if the subsystem is specified correctly. If not, then correct the source or target subsystem parameter and resubmit the job.

CKZ61202E **SUBTASK nn, DB2 SUBSYSTEM ssss NOT DEFINED, CAFRC=hhh**

Explanation:

The CAF return indicates the DB2 subsystem is not defined. nn = subtask number ssss = DB2 subsystem hhh = DB2 call attach facility return code

User response:

Specify the source and target subsystems correctly and resubmit the job.

CKZ61203E **SUBTASK nn, DB2 SUBSYSTEM ssss CAF ERROR USING FUNCTION fff, RC=rrr, RS=sss**

Explanation:

The CAF error return is unexpected. nn = subtask number ssss = DB2 subsystem fff = call attach facility function attempted rrr = call attach facility return code sss = call attach facility reason code

User response:

Contact your system programmer.

CKZ61204I **SUBTASK nn, CONNECT TO DB2 SUBSYSTEM ssss RELEASE rrr**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem rrr = DB2 release number

User response:

No action is required.

CKZ61205E **SUBTASK *subtask_number*, DB2 COMMAND RESPONSE PARSING ERROR, RC=*return_code*, RS=*reason_code***

Explanation:

A parsing error occurred while processing DB2 command output. This is a probable internal error due to unexpected Db2 command output.

User response:

Contact IBM Software Support.

CKZ61206I **SUBTASK nn, DISCONNECT FROM DB2 SUBSYSTEM ssss**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem

User response:

No action is required.

CKZ61207E **SUBTASK nn, OPEN PLAN ppp ON DB2 SUBSYSTEM ssss HAS FAILED, PLAN UNAVAILABLE, RC=rrr**

Explanation:

The CAF return indicates the DB2 open plan failed, as it is not available. nn = subtask number ppp = plan name from PARMLIB ssss = DB2 subsystem rrr = return code

User response:

Bind the correct plan and resubmit the job.

CKZ61208E **SUBTASK nn, SUBSYSTEM ssss CAF REQUEST ERROR, PLAN PPP FUNCTION fff, RC=rrr, RS=sss**

Explanation:

The CAF return indicates the DB2 open plan failed. as it is not available. nn = subtask number ssss = DB2 subsystem ppp = plan name fff = call attach facility function rrr = call attach facility return code sss = call attach facility reason code

User response:

Bind the correct plan and resubmit the job.

CKZ61209I **SUBTASK nn, PLAN ppp OPEN ON DB2 SUBSYSTEM ssss**

Explanation:

This message is informational. nn = subtask number
ppp = plan name ssss = DB2 subsystem

User response:

No action is required.

CKZ61210I **SUBTASK nn, PLAN ppp CLOSE,
DB2 SUBSYSTEM ssss**

Explanation:

This message is informational. nn = subtask number
ppp = plan name ssss = DB2 subsystem

User response:

No action is required.

CKZ61211I **SUBTASK nn, DB2 COMMAND
ISSUED ON ssss: ccc**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem ccc = text of the DB2 command

User response:

No action is required.

CKZ61212I **SUBTASK nn, DB2 COMMAND
REISSUED**

Explanation:

This message is informational. The previous command
had to be reissued as the response area was not large
enough. nn = subtask number

User response:

No action is required.

CKZ61213E **SUBTASK nn, DB2 COMMAND
ERROR, RC=rrr, RS=sss**

Explanation:

The CAF return indicates a secondary error. This error
is returned in field IFCADD. nn = subtask number rrr =
call attach facility return in IFCARC1 sss = call attach
facility reason in IFCARC2

User response:

Contact your systems programmer. See the DB2 Codes
manual for the return and reason code values.

CKZ61214E **SUBTASK nn, DB2 COMMAND
ERROR, NOT AUTHORIZED TO
ISSUE COMMANDS**

Explanation:

The CAF return indicates CKZ can not issue
commands. This error is returned in the IFCADD field
as X'20' nn = subtask number

User response:

Contact your systems programmer.

CKZ61215E **SUBTASK nn, DB2 COMMAND
ERROR, RC=rrr**

Explanation:

The CAF return indicates a secondary error. This error
is returned in field IFCADD. nn = subtask number rrr =
call attach facility return in IFCADD

User response:

Contact your systems programmer. See the DB2 Codes
manual for the return code value.

CKZ61216E **PARSE ERROR: FIRST TICK MUST
BE AT THE START OF A TOKEN...**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ61217E **PARSE ERROR: TWO TICKS IN A
ROW...**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ61218E **PARSE ERROR: EOF
ENCOUNTERED WHILE IN A
TICKED CLAUSE...**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ61219E **PARSE ERROR: ONE TICK BY
ITSELF...**

Explanation:

This is a user error.

User response:

Correct the input and resubmit the job.

CKZ61221I **SUBTASK nn, DB2 SUBSYSTEM
ssss, tt ddd.sss ALL PARTITIONS
ALREADY STOPPED**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name

User response:

No action is required.

CKZ61222I **SUBTASK nn, DB2 SUBSYSTEM
ssss STOP SUCCESSFUL FOR tt
ddd.sss (ALL PARTITIONS)**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name

User response:

No action is required.

CKZ61223I **SUBTASK nn, DB2 CMD RESP ON
ssss, DSN=dsname STAT=SSS**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem sss = status returned from
DISPLAY DB2 command

User response:

No action is required.

CKZ61224I **SUBTASK nn, DB2 SUBSYSTEM
ssss, tt ddd.sss PARTITION ppp,
ALREADY STOPPED**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name
ppp = partition

User response:

No action is required.

CKZ61225I **SUBTASK nn, DB2 SUBSYSTEM
ssss, tt ddd.sss ALREADY
STOPPED**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name

User response:

No action is required.

CKZ61226E **SUBTASK nn, DB2 SUBSYSTEM
ssss, tt ddd.sss PARTITION ppp,
UNEXPECTED STATUS uuu, RC=rrr,
RS=aaa**

Explanation:

This message indicates a status other than STOP,
STOPP, RO and RW was received. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name
ppp = partition uuu = unexpected status value rrr =
return code aaa = reason code

User response:

Change the status of the space(s) and resubmit the job
to copy every space selected for copy. To just copy the
spaces with the desired status, change the parms such
that MAX_COPY_RC=8 and MAX_RC=4. All spaces not
copied will have a copy RC=8 in the Status Report at
job end.

CKZ61227E **SUBTASK nn, DB2 SUBSYSTEM
ssss, tt ddd.sss UNEXPECTED
STATUS uuu, RC=rrr, RS=aaa**

Explanation:

This message indicates a status other than STOP,
STOPP, RO or RW was received. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name
uuu = unexpected status value rrr = return code aaa =
reason code This could happen if for example if a LOB
table space is in ICOPY status.

User response:

Change the status of the space(s) and resubmit the job
to copy every space selected for copy. To just copy the
spaces with the desired status, change the parms such
that MAX_COPY_RC=8 and MAX_RC=4. All spaces not
copied will have a copy RC=8 in the Status Report at
job end.

CKZ61228E **SUBTASK nn, DB2 COMMAND
RESPONSE, RC=rrr, RS=aaa**

Explanation:

This message indicates an unexpected status was
received. See CKZ61229E for the response. nn =
subtask number rrr = return code aaa = reason code

User response:

Change the status of the space and resubmit the job.

CKZ61229E **SUBTASK nn, rrr**

Explanation:

This message is paired with CKZ61228E and prints the
response from DB2. nn = subtask number rrr = DB2
response

User response:

Change the status of the space and resubmit the job.

CKZ61230I **SUBTASK nn, DB2 SUBSYSTEM
ssss STOP SUCCESSFUL FOR tt
ddd.sss PARTITION ppp**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name
ppp = partition

User response:

No action is required.

CKZ61231I **SUBTASK nn, DB2 SUBSYSTEM
ssss STOP SUCCESSFUL FOR tt
ddd.sss**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name

User response:

No action is required.

CKZ61234E **SUBTASK nn, DB2 COMMAND
RESPONSE, EXPECTING eee, NOT
FOUND, RC=rrr RS=sss**

Explanation:

This message indicates the expected status was not received. nn = subtask number eee = response expected rrr = return code sss = reason code

User response:

Change the status of the space and resubmit the job.

**CKZ61235E SUBTASK nn, DB2 SUBSYSTEM
 ssss STOP UNSUCCESSFUL FOR tt
 ddd.sss PARTITION ppp**

Explanation:

This message is an error. An unexpected status was returned to the DB2 display command. This data set will not be processed further. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response:

Change the status of the space and resubmit the job.

**CKZ61236E SUBTASK nn, DB2 SUBSYSTEM
 ssss STOP UNSUCCESSFUL FOR tt
 ddd.sss**

Explanation:

This message is an error. An unexpected status was returned to the DB2 display command. This data set will not be processed further. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response:

Change the status of the space and resubmit the job.

**CKZ61237I SUBTASK nn, DB2 SUBSYSTEM
 ssss START NOT REQUIRED DUE
 TO INITIAL STOP STATUS FOR tt
 ddd.sss PARTITION ppp**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response:

No action is required.

**CKZ61238I SUBTASK nn, DB2 SUBSYSTEM
 ssss START NOT REQUIRED DUE
 TO INITIAL STOP STATUS FOR tt
 ddd.sss**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response:

No action is required.

**CKZ61241I SUBTASK nn, DB2 SUBSYSTEM
 ssss, tt ddd.sss ALL PARTITIONS
 ALREADY STARTED OR NOT BEING
 STARTED**

Explanation:

This message is informational. Note that "Not being Started" refers to those spaces that were in STOP or STOPP status when the initial display was output. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response:

No action is required.

**CKZ61242I SUBTASK nn, DB2 SUBSYSTEM
 ssss START SUCCESSFUL FOR tt
 ddd.sss (ALL PARTITIONS)**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response:

No action is required.

**CKZ61243I SUBTASK nn, DB2 SUBSYSTEM
 ssss INITIALLY STOPPED, tt
 ddd.sss NO START REQUIRED (ALL
 PARTITIONS)**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response:

No action is required.

**CKZ61248I SUBTASK nn, DB2 SUBSYSTEM
 ssss START SUCCESSFUL FOR tt
 ddd.sss PARTITION ppp**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response:

No action is required.

**CKZ61249I SUBTASK nn, DB2 SUBSYSTEM
 ssss START SUCCESSFUL FOR tt
 ddd.sss**

Explanation:

This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response:

No action is required.

CKZ61253E **SUBTASK nn, DB2 SUBSYSTEM
ssss START UNSUCCESSFUL FOR tt
ddd.sss PARTITION ppp**

Explanation:

This message is informational. nn = subtask number
ssss = DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name
ppp = partition

User response:

Contact your systems programmer.

CKZ61254E **SUBTASK nn, DB2 SUBSYSTEM
ssss START UNSUCCESSFUL FOR tt
ddd.sss**

Explanation:

This may be a DB2 problem. nn = subtask number ssss =
DB2 subsystem tt = space type (table space or
indexspace) ddd = database name sss = space name

User response:

Contact your systems programmer.

CKZ61255E **SUBTASK nn, DB2 SUBSYSTEM
ssss, tt ddd.sss BAD PARTITION
NUMBER, EXPECTING nnn,
RECEIVED ooo**

Explanation:

This may be a catalog consistency problem. nn =
subtask number ssss = DB2 subsystem tt = space type
(table space or indexspace) ddd = database name sss
= space name nnn = partition number ooo = partition
number

User response:

Resolve the inconsistency and resubmit the job.

CKZ61261I **SUBTASK nn, WAITING FOR ONE
OR MORE OBJECT STATE
CHANGES, TIME REMAINING TO
WAIT = ddd SECONDS**

Explanation:

This message is informational. nn = subtask number
ddd = decimal number

User response:

No action is required.

CKZ61262E **SUBTASK nn, DB2 SUBSYSTEM
ssss ccc FAILED FOR tt ddd.sss
STATUS=ttt, EXPECTED
STATUS=eee, RC=rrr, RS=zzz**

Explanation:

This message indicates a status other than the desired
status was received. nn = subtask number ssss = DB2
subsystem ccc = DB2 command tt = space type (table
space or indexspace) ddd = database name sss =

space name ttt = current status value eee = expected
status value rrr = return code zzz = reason code

User response:

Change the status of the space and resubmit the job.

CKZ61263I **SUBTASK nn, DB2 SUBSYSTEM
ssss ccc PENDING FOR tt ddd.sss
STATUS=ttt, EXPECTED
STATUS=zzz**

Explanation:

This message indicates a status other than the desired
status was received. CKZ continues to attempt the
change. nn = subtask number ssss = DB2 subsystem
ccc = DB2 command tt = space type (table space or
indexspace) ddd = database name sss = space name
ttt = current status value zzz = expected status value

User response:

No action is required.

CKZ61264I **Subtask *subtask_number*,
*response_line***

Explanation:

This message is informational.

User response:

No action is required.

CKZ61266E **Subtask *subtask_number*, DB2
command error,
DSNTnnnI_message.**

Explanation:

A DB2 command response resulted in an error that
must be resolved before continuing. For instance, an
authorization failure may have occurred that inhibits
further processing by Db2 Cloning Tool Table Space
Cloning. The specific problem can be identified in the
message output, found in the CKZLOG file.

User response:

Locate and review the information for the DSNTnnnI
message in the DB2 for z/OS Messages
documentation. Take the specified corrective action
and resubmit the job.

CKZ61267I **Subtask *subtask_number*, unable
to connect to SSID *subsystem_ID*
using CAF, plan *plan_name*,
function call *attach_facility*,
RC=*return_code*, RS=*reason_code***

Explanation:

This message is informational.

User response:

No action is required.

CKZ61321E **SUBTASK nn, FATAL SQL ERROR
HAS OCCURRED**

Explanation:

This error terminates processing. nn = subtask number

User response:

See the detailed message with the DB2 SQL error.

CKZ61322I **SUBTASK nn, SQL RETRY
ATTEMPT IN PROGRESS, ddd
RETRY(S) REMAINING**

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ61324E **SUBTASK nn, SQL RETRY
ATTEMPTS TERMINATED**

Explanation:

This error terminates processing. nn = subtask number

User response:

See the detailed message with the DB2 SQL error.

CKZ61325W **SQL ERROR IN MODULE
module_name**

Explanation:

This is a warning message. The printed SQL error follows this message.

User response:

See the detailed message with the Db2 SQL error.

CKZ61326E **SUBTASK nn, DSNTIAR ERROR
PRINTING SQL ERROR ON DB2
SUBSYSTEM ssss, RC=rrr**

Explanation:

The SQL error cannot be printed by DB2 module DSNTIAR. nn = subtask number ssss = DB2 subsystem rrr = R15 return from DSNTIAR

User response:

Contact your system programmer.

CKZ61327I **Subtask *subtask_number*, Object
object_type
object_creator.object_name is
implicit, matching target object
will be found**

Explanation:

This message is informational.

User response:

No action is required.

CKZ61328W **Subtask *subtask_number*,
matching target object for
object_type
object_creator.object_name is not
found and object-translate is not
specified. Copy may result in error**

Explanation:

The source object listed in the message could not be matched with a source object. Possible reasons are that the target object does not exist; or object-translate is required.

User response:

Investigate the potential cause based on the suggestions above. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ61329I **Subtask *subtask_number*,
matching target object for
object_type
object_creator.object_name is not
found. Specified object-translate
will be used**

Explanation:

This message is informational.

User response:

No action is required.

CKZ61330I **Subtask *subtask_number*, implicit
source object *object_type*
source_object_creator.source_object
_name is paired with target object
target_object_creator.target_object
*_name***

Explanation:

This message is informational.

User response:

No action is required.

CKZ61331W **Subtask *subtask_number*, implicit
object *object_type*
database_name.space_name will
not be copied because its base
object was not included in
LISTDEF**

Explanation:

The source object that is listed in the message could not be matched with any target object, because base objects are required for matching implicit objects.

User response:

Change the LISTDEF to include the base object, or specify OBJECT-TRANSLATE for implicit objects.

CKZ61332W **Subtask *subtask_number*, Explicit
object *object_type*
database_name.space_name |
object_creator.object_name will not
be copied because its base object
was not included in LISTDEF**

Explanation:

The explicitly created object that is listed in the message could not be copied because its base object was not included in the LISTDEF. Base objects are required for matching explicit objects.

User response:

Change the LISTDEF to include the base object.

CKZ61333I Subtask *subtask_number*, Auto translation used for object *object_type* *database_name.space_name* | *object_creator.object_name*, matching target object will be found

Explanation:

The AUTO-TABLESPACE-TRANSLATE(Y) or the AUTO-INDEXSPACE-TRANSLATE(Y) keywords or both were specified. Db2 Cloning Tool will locate matching target objects.

User response:

No response is required.

CKZ61334I Subtask *subtask_number*, Explicit source object *object_type* *source_database_name.source_space_name* | *source_object_creator.source_object_name* is paired with target object *target_database_name.target_space_name* | *target_object_creator.target_object_name* by auto translate

Explanation:

The AUTO-TABLESPACE-TRANSLATE(Y) or the AUTO-INDEXSPACE-TRANSLATE(Y) keyword was specified. Db2 Cloning Tool located the matching target object for the explicit source object that is listed in the message.

User response:

No response is required.

CKZ61335E Subtask *subtask_number*, For source object **IX** *index_creator.index_name*, more than one target object found. Use explicit **OBJECT-TRANSLATE** statement to match this source object with target object.

Explanation:

The AUTO-INDEXSPACE-TRANSLATE(Y) keyword was specified. For the source index, Db2 Cloning Tool found more than one target index with the same properties. The indexes cannot be automatically matched.

User response:

Specify the **OBJECT-TRANSLATE** statement to match the source index to the correct target index and resubmit the job.

CKZ61336W Subtask *subtask_number*, Implicit object *object_type object_name*

will not be copied because *object_type object_name* was excluded from copying

Explanation:

This message is a warning that an implicit object will not be copied.

User response:

Review other associated messages to determine the cause and fix the warning.

CKZ61337W Subtask *subtask_number*, Explicit object *object_type object_name* will not be copied because *object_type object_name* was excluded from copying

Explanation:

This message is a warning that an explicit object will not be copied.

User response:

Review other associated messages to determine the cause and fix the warning.

CKZ61338E Subtask *subtask_number*, Matching target object *target_database_name.target_space_name* for source object *object_type* *source_database_name.source_space_name* is found, but it resides in an unexpected database. Expected database: *database_name*, database from Db2 catalog: *database_name*

Explanation:

There are inconsistencies in the object translate rules between the database name, object creator, and object name. The matching target space for an implicitly created source space is found through its base object (using object translate rules for object creator and object name), but the database name of the matching target object conflicts with the object translate rules for the database.

User response:

Correct the object translate rules and resubmit the job.

CKZ61901I Unmatched index *creator.ixname* found on table *creator.tbname*

Explanation:

A target-only index was found on a table. This index will be rebuilt if **REBUILD-UNMATCHED-TARGET-INDEXES(Y)** is specified.

User response:

No action is required.

CKZ64000W **SUBTASK** *subtask_number*, **UNABLE TO READ TARGET**

**CATALOG, RC=return_code,
RS=reason_code
OBJECT=object_type
object_creator.object_name**

Explanation:

This may be a DB2 SQL error.

User response:

Correct the DB2 problem and resubmit the job.

CKZ64001E Subtask *subtask_number*, Target subsystem *subsystem*, Object was not found in cache, *object_type object_qualifier_1.object_qualifier_2*

Explanation

The object that is listed in the message was not found in the cache for one of the following reasons:

- The cache was not populated on the target system.
- The cache is not consistent with the current catalog state.
- The database in which the object should be found was not specified in TARGET-PREFETCH-DATABASE-LIST. Therefore, the database and its child objects were not cached.
- The object does not exist.

User response:

Run the job with ENABLE-TARGET-PREFETCH(Y) to populate or refresh the cache on the target system for the required databases.

CKZ64002E Subtask *subtask_number*, Target subsystem *subsystem*, Database *database_name* was not found in cache

Explanation

The object that is listed in the message was not found in the cache for one of the following reasons:

- The cache was not populated on the target system.
- The cache is not consistent with the current catalog state.
- The database was not specified in TARGET-PREFETCH-DATABASE-LIST. Therefore, the database and its child objects were not cached.
- The object does not exist.

User response:

Run the job with ENABLE-TARGET-PREFETCH(Y) to populate or refresh the cache on the target system for the required databases.

CKZ64003E Subtask *subtask_number*, Target subsystem *subsystem*, No

child_object_type was/were found in cache for *parent_object_type object_qualifier_1.object_qualifier_2*

Explanation

The object that is listed in the message was not found in the cache for one of the following reasons:

- The cache was not populated on the target system.
- The cache is not consistent with the current catalog state.
- The database in which object should be found was not specified in TARGET-PREFETCH-DATABASE-LIST. Therefore, the database and its child objects were not cached.
- The object does not exist.

User response:

Run the job with ENABLE-TARGET-PREFETCH(Y) to populate or refresh the cache on the target system for the required databases.

CKZ64004E Subtask *subtask_number*, Target subsystem *subsystem*, No *child_object_type* was/were found in catalog for *parent_object_type object_qualifier_1.object_qualifier_2*

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ64005E Subtask *subtask_number*, Target subsystem *subsystem*, Failed to reread catalog for *object_type object_qualifier_1.object_qualifier_2*

Explanation:

A catalog inconsistency was encountered by Db2 Cloning Tool. This is probably a Db2 error.

User response:

Resolve the inconsistency and resubmit the job.

CKZ64006W Subtask *subtask_number*, Target subsystem *subsystem*, No tables found in target table space *database_name.table_space_name*

Explanation:

No table objects were found in the target table space.

User response:

Add the table objects and resubmit the job.

CKZ64007E Subtask *subtask_number*, Target subsystem *subsystem*, IX *index_creator.index_name* resides in unexpected database. Expected database: *expected_database*, database from Db2 catalog: *database_from_catalog*

Explanation:

The OBJECT-TRANSLATE rules for DATABASE indicate that the specified index should reside in *expected_database*, but index information from the Db2 catalog indicates that it resides in *database_from_catalog*.

User response:

Correct the OBJECT-TRANSLATE rules for CREATOR, INDEX, and DATABASE and resubmit the job.

CKZ06574E FCTOXRCP PRIMARY KEYWORD IS NOT COMPATIBLE WITH FASTREP(NONE)

Explanation:

The FASTREP parameter must be set to PREF or REQ to use FCTOXRCP PRIMARY.

User response:

Correct the input and resubmit the job.

CKZ66901E Subtask *subtask_number*, ACK expected from *remote_DB2_subsystem*, receive type *receive_message_type*

Explanation:

An ACK was expected from the remote TCP/IP connection for the specified receive type.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ66902E Block format error, *ttnn,parameter1,parameter2*

Explanation

There is an error in the packet received from the remote Db2 subsystem. *ttnn* indicates the error type and a decimal qualifier. Possible *tt* values include:

- BL: Block number receive error
- CK: TCP/IP packet checksum error
- GM: Group accessed by minilog not found
- MM: Space group cannot be found
- RM: Number of records does not match array length
- UN: Unknown block type received
- UT: Unexpected type from server
- XB: No blocks found

- XG: Group block should first from server

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ66903I *message_explanation, number_of_records*

Explanation:

This message indicates that the process was successful.

User response:

No action is required.

CKZ66990I *group_ID, number_of_bytes_in_hex*

Explanation:

This informational message indicates that the process was successful.

User response:

No action is required.

CKZ66994I Processing group *group_ID*

Explanation:

This informational message indicates that process for the group that is listed in the message has begun.

User response:

No action is required.

CKZ67101I CKZZ output open OK for DDNAME *ddname*

Explanation:

This message is informational. CKZZ is the SYSINCKZ LOG-APPLY work data set.

User response:

No action is required.

CKZ67102W CKZZ output open failed for DDNAME *ddname*, RC=*return_code*, RS=*reason_code*

Explanation:

This message is a warning that the SYSINCKZ could not be opened.

User response:

Correct the CKZZ data set and resubmit the job.

CKZ67103I Target server cache populating was completed with RC=*return_code*

Explanation:

Target server cache population completed with the specified return code.

User response:

If the return code is not zero, the target server cache will be cleaned up. Check server messages for errors,

resolve the problems described in these messages, and rerun the source job to repopulate the cache.

CKZ67104I **Target server cache was cleaned up**

Explanation:

The target server cache was cleaned up.

User response:

If the server cache is no longer needed, no action is required. If you want the target cache to be populated and used again, run the source job with ENABLE-TARGET=PREFETCH(Y).

CKZ67111I **TCPIP Client Disconnected, Socket=socket_number**

Explanation:

This message is informational. It is issued by the TCP/IP server job.

User response:

No action is required.

CKZ67140E **TCPIP server receive block error, fff, received rrr expecting xxx**

Explanation:

This message indicates an error occurred with the TCP/IP block number received from the client. This may be a TCP/IP transmission error or an internal programming error. fff = the function being performed rrr = block number received xxx = block number expecting

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ67145E **TCPIP server received block with a format error processing log pages, reason_code**

Explanation:

An invalid TCP/IP header was received. The connection has been shut down.

User response:

If unable to resolve the error, contact IBM Software Support.

CKZ67146I **number_of_lines XMLSTRINGS lines read from source catalog**

Explanation:

This message is informational.

User response:

No action is required.

CKZ67147E **ddname DD open error, RC=return_code, RS= reason_code**

Explanation:

This message is an error that the DD for DDL could not be opened.

User response:

Correct the reason for the error and resubmit the job. If unable to resolve this error, contact IBM Software Support.

CKZ67200E **UNABLE TO ACCEPT CONNECTION WITH PORT tcpip_bound_port, RETCODE=return_code, ERRNO=tcpip_reason_code**

Explanation:

This message indicates a probable TCP/IP error.

User response:

Contact your system programmer.

CKZ67201E **Unable to change socket options: RC=return_code, RS=reason_code**

Explanation:

This message indicates a probable TCP/IP error. This might be an environment error or an internal programming error.

User response:

Contact your systems programmer. If unable to resolve this error, contact IBM Software Support.

CKZ67202E **Unable to resolve DNS name name: RC=return_code, RS=reason_code**

Explanation:

This message indicates a user error or a TCP/IP environment error.

User response:

Correct the DNS name and resubmit the job. If unable to resolve this error, contact your systems programmer or IBM Software Support.

CKZ67203I **DNS name name was resolved to IP ip**

Explanation:

The IP address *ip* corresponds to DNS name *name*.

User response:

No action is required.

CKZ67204E **Unable to connect to any resolved address of name**

Explanation:

The client was unable to connect to the server job using the specified DNS name. This message can indicate a user error or a TCP/IP environment error.

User response:

Verify that a TCP/IP server job is waiting for a connect at the DNS name and port that is specified in the source job. If unable to make the connection, contact your systems programmer.

CKZ67205E **FREEADDRINFO error:**
RC=return_code, RS=reason_code

Explanation:

This message indicates an internal programming error.

User response:

Contact IBM Software Support.

CKZ67206I **CONNECTION ATTEMPT TO IP**
server_ipaddress, PORT
TCPIP_bound_port

Explanation:

This message is informational. It is issued by the TCPIP client job.

User response:

No action is required.

CKZ67207I **TCPIP CONNECTION**
ESTABLISHED ON PORT
TCPIP_bound_port USING IP
server_ipaddress

Explanation:

This message is informational. It is issued by the TCPIP client job.

User response:

No action is required.

CKZ67208I **TCPIP CONNECTION, IP**
ip_address, SOCKET
tcpip_socket_number

Explanation:

This message is informational. It is issued by the TCP/IP client job.

User response:

No action is required.

CKZ67209E **CONNECT ERROR FOR IP**
ip_address, RETCODE=return_code,
ERRNO=tcpip_reason_code

Explanation:

This message indicates a user error or a TCP/IP environment error.

User response:

Correct the IP address and resubmit the job or contact your systems programmer.

CKZ67210E **Connect timed out for IP ipaddr,**
retries exhausted,
RETCODE=return_code,
ERRNO=tcpip_reason_code

Explanation:

This message indicates that the source job is unable to connect to the TCP/IP server job.

User response:

Verify that a TCP/IP server job is waiting for a connect at the IP address and port that is specified in the source job. If unable to make the connection, contact your systems programmer.

CKZ67211I **BIND CALL SUCCESSFUL, PORT**
tcpip_bound_port

Explanation:

This message is informational. It is issued by the TCP/IP server job.

User response:

No action is required.

CKZ67212I **LISTEN CALL SUCCESSFUL, PORT**
tcpip_bound_port

Explanation:

This message is informational. It is issued by the TCP/IP server job.

User response:

No action is required.

CKZ67213I **TCPIP WAITING FOR CLIENT**
CONNECT ON SERVER PORT
tcpip_port_waiting_for_connect

Explanation:

This message is informational. It is issued by the TCP/IP server job after initialization and after a client disconnect.

User response:

No action is required.

CKZ67214I **TCPIP CLIENT CONNECT FROM IP**
ip_address

Explanation:

This message is informational. It is issued by the TCP/IP server job.

User response:

No action is required.

CKZ67215I **ACCEPT CALL SUCCESSFUL, IP**
ip_address, NEW SOCKET
socket_allocated_for_this_connect

Explanation:

This message is informational. It is issued by the TCP/IP server job.

User response:

No action is required.

CKZ67216E **UNABLE TO LISTEN ON PORT**
tcpip_bound_port,
RETCODE=return_code,
ERRNO=tcpip_reason_code

Explanation:

This message indicates a user error or a TCP/IP environment error.

User response:

Correct the port number and resubmit the job or contact your systems programmer.

CKZ67217I TCPIP SERVER CONNECTION NOT AVAILABLE, RETRY IN ONE MINUTE, *decimal_number* RETRIES REMAINING

Explanation:

This message is informational. It is issued by the TCP/IP client job.

User response:

No action is required.

CKZ67218E UNABLE TO LOAD TCP Module FROM EZA LIBS TO EXECUTE EZASMI_macro_that_failed EZASMI MACRO

Explanation:

This is a TCP/IP environment error. Cannot load the TCP/IP module EZASOH03.

User response:

Contact your system programmer to give Db2 Cloning Tool Table Space Cloning access to the TCP/IP load library SEZALOAD.

CKZ67219E UNABLE TO ALLOCATE SOCKET, RETCODE=*return_code*, ERRNO=*tcpip_ERRNO_return_value*

Explanation:

This is a TCP/IP environment error.

User response:

Contact your system programmer.

CKZ67220E UNABLE TO BIND TO SOCKET WITH PORT *tcpip_port_number*, RETCODE=*return_code*, ERRNO=*tcpip_ERRNO_return_value*

Explanation:

This is a TCP/IP environment error. The port specified may be in use by another application.

User response:

Specify another port number. Ensure that the port number of the TCP/IP client source job matches the port number of the TCP/IP server job.

CKZ67221E CLOSE ERROR, RETCODE=*return_code*, ERRNO=*tcpip_ERRNO_return_value*

Explanation:

This is a probable TCP/IP error.

User response:

Contact your system programmer.

CKZ67222E UNABLE TO CONNECT TO TCP *name_of_tcpip_address_space*,

RETCODE=*return_code*,

ERRNO=*tcpip_ERRNO_return_value*

Explanation:

This is a user error or a TCP/IP error.

User response:

Specify the correct TCP/IP started task name and restart the job. If not resolved, contact your system programmer.

CKZ67223E RECEIVE DATA LENGTH ERROR, EXPECTING *number_of_bytes_expected*, RECEIVED *number_of_bytes_received*

Explanation:

This is a probable TCP/IP error.

User response:

Contact your system programmer.

CKZ67224E RECEIVE DATA ERROR, RETCODE=*return_code*, ERRNO=*tcpip_ERRNO_return_value*

Explanation:

This is a probable TCP/IP error.

User response:

Contact your system programmer.

CKZ67225E SEND DATA ERROR, RETCODE=*return_code*, ERRNO=*tcpip_ERRNO_return_value*

Explanation:

This is a probable TCP/IP error.

User response:

Contact your system programmer.

CKZ67226I SOCKET CALL SUCCESSFUL, SOCKET *socket_number*

Explanation:

This message is informational.

User response:

No action is required.

CKZ67227I TCPIP SOCKET CLOSED, IP *ipaddress*, - SOCKET *socket_number*, PACKETS RECEIVED *decimal_number*, PACKETS SENT *decimal_number*

Explanation:

This message is informational.

User response:

No action is required.

CKZ67228I INITAPI CALL SUCCESSFUL, TCP *started_task_of_tcpip*

Explanation:

This message is informational.

User response:

No action is required.

CKZ67229I RECEIVE
decimal_number_of_bytes_
received_without_error BYTES, OK

Explanation:

This message is informational.

User response:

No action is required.

CKZ67230I SEND
decimal_number_of_bytes_sent_wi
thout_error BYTES, OK

Explanation:

This message is informational.

User response:

No action is required.

CKZ67231E Unable to set up network
encryption: reason

Explanation

This is a user error. This message indicates that TCP/IP communications encryption could not be set up due to one of the specified reasons:

- The server is set up to use encryption, but the client (the source or target Db2 Cloning Tool job) is not set up for encryption, or vice versa.
- The server and client are configured to use different encryption keys.

User response:

See the description of the TCPIP_KEY_LABEL and the TCPIP_ENCRYPTION_ENABLE CKZINI keywords for information on how to set up TCP/IP encryption. Correct the error and resubmit the job.

CKZ67232I Did not receive the response from
the server. Will wait for
time_in_minutes more minutes

Explanation:

This informational message is issued by the TCP/IP client job.

User response:

No action is required.

CKZ67233E Receive timed out for connection
to IP ip_address

Explanation:

This message indicates that the TCP/IP connection was successfully established, but the server did not respond to a request. This error might occur when the

TCP/IP port is in use by another application, or when the server has more than one connected client and the server is servicing another client.

User response:

Investigate which application is using the port. If the port is being used by another application, release the port and start the server. If the server is running, check for other connected clients.

CKZ67234E Unable to set socket options,
RETcode=return_code,
ERRNO=tcpip_ERRNO_return_value

Explanation:

This is a probable TCP/IP error.

User response:

Contact your systems programmer.

CKZ67235E Network protocol version
mismatch. Server protocol level:
server_level, client protocol level:
client_level

Explanation:

This message indicates that maintenance levels on the Db2 Cloning Tool client and server are not compatible. *server_level* and *client_level* indicate the maintenance level of the Db2 Cloning Tool server and client respectively.

User response:

Update the Db2 Cloning Tool client or server libraries and resubmit the job.

CKZ6S051E Parameter file data set does not
exist

Explanation:

The CPARM parameter file is required.

User response:

Specify the valid path for the CPARM parameter file and resubmit the job.

CKZ6S052E Parameter file data set has invalid
type

Explanation:

The CPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Specify the CPARM parameter file as a file with correct data type and resubmit the job.

CKZ6S053E Failed to determine attributes of
parameter file data set

Explanation:

Validation of CPARM file attributes failed. The CPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Ensure that the CPARM parameter file is specified with the correct attributes. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S054E Parameter file data set has invalid record format or record length

Explanation:

The CPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Specify the CPARM parameter file as a file with correct attributes and resubmit the job.

CKZ6S055E An error occurred while reading the parameter file (line n): system_message

Explanation:

The utility could not read from the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S056E Invalid parameter file (line n): Incorrect parameter definition

Explanation:

An error occurred due to incorrect input data in the CPARM parameter file.

User response:

Check the documentation for the parameter file syntax and correct the syntax before resubmitting the job. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S057E An error occurred while writing to the parameter file: system_message

Explanation:

The utility could not write to the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S058E Cannot open parameter file: DSN = 'data_set_name'[Member = 'member']. system_message

Explanation:

The utility could not open the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S059E Cannot open parameter file: DDN = dd_name. system_message

Explanation:

The utility could not open the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S081E Unable to get current user ID

Explanation:

The utility could not retrieve information about the current user ID.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S082E Unable to get current time

Explanation:

The utility could not retrieve information about the current time.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S101E The data set is not a profile repository: HLQ = 'repository_hlq'

Explanation:

An invalid HLQ was specified for the profile repository.

User response:

Specify a valid profile repository and resubmit the job.

CKZ6S102W The length of parameter value exceeds the maximum length. Value of parameter parameter_name will be truncated

Explanation:

The parameter value that is listed in the message is too long. The value will be truncated to the allowed length.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S103E The first parameter in the CPARM file must be the PROFILE-NAME

Explanation:

An error occurred due to incorrect input data in the CPARM parameter file.

User response:

Ensure that the CPARM parameter file was specified. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S104I **Import of subsystem profile
'profile_creator':profile_name'
started | Original profile was
'original_profile_creator':profile_name'**

Explanation:

The message marks the beginning of import of the subsystem cloning profile that is listed in the message.

User response:

No action is required.

CKZ6S105I **Import of subsystem profile
'profile_creator':profile_name'
completed successfully. { |
Original profile name was:
'profile_name' | Profile was
skipped}**

Explanation:

The subsystem profile that is listed in the message has been successfully imported.

User response:

No action is required.

CKZ6S106E **An internal error occurred while
working with profile repository:
system_message**

Explanation:

An error occurred.

User response:

Ensure that correct and latest versions of all required load libraries are used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6S107W **The list of clone profiles for export
is empty**

Explanation:

The export of clone profiles was launched, but no profiles were found for export.

User response:

Check the parameters of the export command and correct if necessary, then resubmit the job.

CKZ6S108I **Export of subsystem profile
'profile_creator':profile_name'
started**

Explanation:

The message marks the beginning of export of the subsystem profile that is listed in the message.

User response:

No action is required.

CKZ6S109W **The length of parameter value
parameter_name exceeds the
maximum length**

Explanation:

The parameter value that is listed in the message is too long.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S110W **Export of subsystem profile
'profile_creator':profile_name'
failed. {No access to profile |
Profile not found}**

Explanation:

The subsystem profile that is listed in the message cannot be exported.

User response:

Ensure that the correct profile repository was used, and that you have sufficient authority to read the profile.

CKZ6S111I **Export of subsystem profile
'profile_creator':profile_name'
completed successfully**

Explanation:

The subsystem profile that is listed in the message has been successfully exported.

User response:

No action is required.

CKZ6S112I **The new name for imported
subsystem profile
'profile_creator':profile_name' will
be: 'profile_name'**

Explanation:

The profile that is listed in the message will be imported with a new name because there is an existing subsystem profile with the same name.

User response:

No action is required.

CKZ6S113I **The subsystem profile
'profile_creator':profile_name'
already exists and will be skipped**

Explanation:

The profile that is listed in the message will not be imported because there is an existing subsystem profile with the same name.

User response:

No action is required.

CKZ6S114I **The subsystem profile '*profile_creator*':*profile_name*' already exists and will be overwritten**

Explanation:

The profile that is listed in the message already exists in the profile repository. The profile will be overwritten.

User response:

No action is required.

CKZ6S115W **The parameter *parameter_name* has incorrect value. Parameter will be initialized by default: '*default_value*'**

Explanation:

The CPARM parameter file contains outdated, unsupported, or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S116W **The required parameter *parameter_name* is not specified**

Explanation:

The CPARM parameter file contains either outdated or incorrect information.

User response:

Ensure that correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S117E **Import of subsystem profile '*profile_creator*':*profile_name*' failed. {Invalid values | No access | The CPARM file does not contain correct values of some required parameters | Profile name exceeds the maximum length}**

Explanation:

The subsystem profile that is listed in the message has not been imported.

User response:

Check previous warnings for more information.

CKZ6S118W **Unknown parameter '*unknown_parameter*' will be skipped**

Explanation:

The CPARM file contain unknown parameters.

User response:

Ensure that correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S119W **Not specified required parameter *parameter_name* will be initialized by default: '*default_value*'**

Explanation:

The CPARM parameter file contains either outdated or incorrect information.

User response:

Ensure that correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S120W **Incompatible values of parameters: *parameter_name* and *parameter_name***

Explanation:

The CPARM parameter file contains either outdated or incorrect information.

User response:

Ensure that correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S121W **Import of subsystem profile '*profile_creator*':*profile_name*' completed with warnings. { | Original profile name was: '*profile_name*' | Profile was skipped}**

Explanation:

The subsystem profile that is listed in the message has been imported. Warnings were encountered during the import.

User response:

Check previous messages for more information.

CKZ6S122E **Export of subsystem profile '*profile_creator*':*profile_name*' failed. Either profile name or profile creator exceeds the maximum length**

Explanation:

An error occurred due to incorrect input data.

User response:

Check previous warnings for more information.

CKZ6S125I **Temporary profile was created**

Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is

deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

CKZ6S126I Temporary profile was deleted

Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

CKZ6S127I Original profile was deleted

Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

**CKZ6S128I Temporary profile renamed to
 { original | new } profile name**

Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

**CKZ6S129I Simulate mode is enabled. No
 changes will be added into the
 profile repository**

Explanation:

This message is informational.

User response:

No action is required.

**CKZ6S186W A different number of source and
 target items in *parameter_name*
 specified**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6S187W Invalid source value for the
 {RENAME-MASKS | DATACLAS-
 PAIRS | MGMTCLAS-PAIRS |
 STORCLAS-PAIRS} specified**

Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6S188W Invalid target value for the
 {RENAME-MASKS | DATACLAS-
 PAIRS | MGMTCLAS-PAIRS |
 STORCLAS-PAIRS} specified**

Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6S191W Too many values for parameter
 {DB2-GROUP-PAIR | DB2-
 SUBSYSTEM-PAIR}. Only the first
 two values will be used**

Explanation:

The CPARM parameter file contain too many values for the parameter that is listed in the message.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6S192W Source value for the parameter
 {DB2-GROUP-PAIR | DB2-
 SUBSYSTEM-PAIR} is not specified**

Explanation:

The source value for the parameter that is listed in the message is required if the parameter is specified in the CPARM file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6S193W Target value for the parameter
 {DB2-GROUP-PAIR | DB2-
 SUBSYSTEM-PAIR} is not specified**

Explanation:

Target value for the parameter that is listed in the message is required if the parameter specified in the CPARM file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S196W **A different number of source and target items in SSID-PAIRS parameter specified**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S197W **A wrong number of items in MEMBER-NAMES parameters specified**

Explanation:

The number of source and target items in the MEMBER-NAMES parameters must be the same and equal to the number of source and target SSIDs.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S198W **SSID 'ssid' specified in the SSID-SURVIVING parameter was not found among target SSIDs**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S199W **Too many values for parameter {SSID-SURVIVING | SURVIVING-PRIMARY-LIST | SURVIVING-DDF-LIST | DDF-DYNAMIC-ALIAS}. Only the necessary values will be used**

Explanation:

The CPARM parameter file contain too many values for the parameter that is listed in the message.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S200W **The surviving target SSID ssid is already defined**

Explanation:

The parameter that is listed in the message was already defined.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S201W **The primary value for surviving target SSID ssid is already defined. New value will be skipped**

Explanation:

The parameter that is listed in the message was already defined.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S202W **The DDF value for surviving target SSID ssid is already defined. New value will be skipped**

Explanation:

The parameter that is listed in the message was already defined.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S203W **Invalid {source | target} SSID: 'ssid' for the DB2-GROUP-PAIR specified**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S204W **Too many values for parameter SSID-SURVIVING with 'NONDS' data sharing attributes of target**

Explanation:

The CPARM parameter file contains incorrect information.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S251I **The CLONING-TYPE will be ONLINE for all subsystem pairs and group pairs by the reason of DB2SLB**

Explanation:

If the profile has either subsystem pairs or group pairs with an OFFLINE cloning type, it will be changed to ONLINE

User response:

No action is required.

CKZ6S261W **Invalid {source | target} SSID: 'ssid' for the DB2-SUBSYSTEM-PAIR specified**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S266W Invalid data set name was entered for WORK-DS-PREFIX

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S271W Invalid value for the EXCLUDE-SRCNAME-MASKS | DDF-DYNAMIC-ALIAS specified

Explanation:

The CPARM file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S281W Invalid {DDF-PORT | DDF-RESPORT | DDF-SECPORT} was specified for the DDF-SSID: 'ssid'

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S286W The '*DEFAULT' can be used as source value only once for the {DATACLAS-PAIRS | MGMTCLAS-PAIRS | STORCLAS-PAIRS}

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6S303W A different number of list items in FROM-STORAGEGROUP | FROM-VOLSER and TO-STORAGEGROUP | TO-VOLSER

Explanation:

The CPARM parameter file contains incorrect information.

User response:

Change the number of items in the lists to the same count.

CKZ6S304W Lists FROM-STORAGEGROUP | TO-STORAGEGROUP and FROM-VOLSER | TO-VOLSER cannot be empty at the same time

Explanation:

The CPARM parameter file contains incorrect information.

User response:

Change the number of items in at least one of the lists so it is not empty.

CKZ6T000E Profile library API is not initialized

Explanation:

An internal error occurred during profile library API initialization.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ6T051E Parameter file data set does not exist

Explanation:

The CPARM parameter file is required.

User response:

Specify the valid path for the CPARM file and resubmit the job.

CKZ6T052E Parameter file data set has invalid type

Explanation:

The CPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Specify the CPARM file as a file with the correct data type and resubmit the job.

CKZ6T053E Failed to determine attributes of parameter file data set

Explanation:

Validation of CPARM parameter file attributes failed. The CPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Ensure that the CPARM file with correct attributes is specified. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T054E Parameter file data set has invalid record format or record length

Explanation:

The CPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Specify the CPARM parameter file as a file with correct attributes and resubmit the job.

CKZ6T055E **An error occurred while reading the parameter file (line n):**
system_message

Explanation:

The utility could not read from the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T056E **Invalid parameter file (line n):**
Incorrect parameter definition

Explanation:

An error occurred due to incorrect input data in the CPARM parameter file.

User response:

Check the documentation for the parameter file syntax and correct the syntax before resubmitting the job. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T057E **An error occurred while writing to the parameter file:**
system_message

Explanation:

The utility could not write to the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T058E **Cannot open parameter file: DSN =**
'data_set_name'[Member =
'member']. system_message

Explanation:

The utility could not open the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T059E **Cannot open parameter file: DDN =**
dd_name. system_message

Explanation:

The utility could not open the CPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T081E **Unable to get current user ID**

Explanation:

The utility could not retrieve information about the current user ID.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T082E **Unable to get current time**

Explanation:

The utility could not retrieve information about the current time.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T101E **The data set is not a profile**
repository: HLQ = 'repository_hlq'

Explanation:

An invalid HLQ was specified for the profile repository.

User response:

Specify a valid profile repository and resubmit the job.

CKZ6T102W **The length of parameter value**
exceeds the maximum length.
Value of parameter
parameter_name will be truncated

Explanation:

The parameter value that is listed in the message is too long. The value will be truncated to the allowed length.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T103E **The first parameter in the CPARM**
file must be the PROFILE-NAME

Explanation:

An error occurred due to incorrect input data in the CPARM parameter file.

User response:

Ensure that the CPARM file was specified. If you have changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T104I **Import of table space profile 'profile_creator':profile_name' started | Original profile was 'original_profile_creator':profile_name'**

Explanation:

This message marks the beginning of import of the table space cloning profile that is listed in the message.

User response:

No action is required.

CKZ6T105I **Import of table space profile 'profile_creator':profile_name' completed successfully. { | Original profile name was: 'profile_name' | Profile was skipped}**

Explanation:

The table space cloning profile that is listed in the message has been successfully imported.

User response:

No action is required.

CKZ6T106E **An internal error occurred while working with profile repository: system_message**

Explanation:

An error occurred.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T107W **The list of clone profiles for export is empty**

Explanation:

The export of clone profiles was launched, but no profiles were found for export.

User response:

Check the parameters of the export command and correct if necessary, then resubmit the job.

CKZ6T108I **Export of table space profile 'profile_creator':profile_name' started**

Explanation:

This message marks the beginning of export of the table space cloning profile that is listed in the message.

User response:

No action is required.

CKZ6T109W **The length of parameter value parameter_name exceeds the maximum length**

Explanation:

The parameter value that is listed in the message is too long.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T110W **Export of table space profile 'profile_creator':profile_name' failed. {No access to profile | Profile not found}**

Explanation:

The table space cloning profile that is listed in the message cannot be exported.

User response:

Make sure that the correct profile repository was used and that you have read access to the profile.

CKZ6T111I **Export of table space profile 'profile_creator':profile_name' completed successfully**

Explanation:

The table space profile that is listed in the message has been successfully exported.

User response:

No action is required.

CKZ6T112I **The new name for imported table space profile 'profile_creator':profile_name' will be: 'profile_name'**

Explanation:

The profile that is listed in the message will be imported with a new name because there is an existing table space cloning profile with the same name.

User response:

No action is required.

CKZ6T113I **The table space profile 'profile_creator':profile_name' already exists and will be skipped**

Explanation:

The profile that is listed in the message will not be imported because there is an existing table space cloning profile with the same name.

User response:

No action is required.

CKZ6T114I **The table space profile 'profile_creator':profile_name' already exists and will be overwritten**

Explanation:

The profile that is listed in the message already exists in the profile repository. The profile will be overwritten.

User response:

No action is required.

CKZ6T115W **The parameter *parameter_name* has incorrect value. Parameter will be initialized by default: '*default_value*'**

Explanation:

The CPARM parameter file contains outdated, unsupported, or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T116W **The required parameter *parameter_name* is not specified**

Explanation:

The CPARM parameter file contains either outdated or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T117E **Import of table space profile '*profile_creator*':'*profile_name*' failed. {Invalid values | No access | The CPARM file does not contain correct values of some required parameters | Profile name exceeds the maximum length}**

Explanation:

The table space cloning profile that is listed in the message has not been imported.

User response:

Check previous warnings for more information.

CKZ6T118W **Unknown parameter '*unknown_parameter*' will be skipped**

Explanation:

The CPARM parameter file contain unknown parameters.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T119W **Not specified required parameter *parameter_name* will be initialized by default: '*default_value*'**

Explanation:

The CPARM parameter file contains either outdated or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T120W **Incompatible values of parameters: *parameter_name* and *parameter_name***

Explanation:

The CPARM parameter file contains either outdated or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T121W **Import of table space profile '*profile_creator*':'*profile_name*' completed with warnings. { | Original profile name was: '*profile_name*' | Profile was skipped }**

Explanation:

The table space cloning profile that is listed in the message has been imported. Warnings were encountered during the import.

User response:

Check previous messages for more information.

CKZ6T122E **Export of table space profile '*profile_creator*':'*profile_name*' failed. Either profile name or profile creator exceeds the maximum length**

Explanation:

An error occurred due to incorrect input data.

User response:

Check previous warnings for more information.

CKZ6T125I **Temporary profile was created**

Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

CKZ6T126I Temporary profile was deleted
Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

CKZ6T127I Original profile was deleted
Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

**CKZ6T128I Temporary profile renamed to
 { original | new } profile name**
Explanation:

During the import, all data is recorded in a temporary profile. If an error occurs, the temporary profile is deleted. Otherwise, the temporary profile either becomes the original profile (the original profile is deleted) or becomes a new profile.

User response:

No action is required.

**CKZ6T129I Simulate mode is enabled. No
 changes will be added into the
 profile repository**
Explanation:

This message is informational.

User response:

No action is required.

**CKZ6T131W The parameter *parameter_name* is
 already defined. New value will be
 skipped**
Explanation:

The parameter name that is listed in the message has already been defined. The new value will be ignored.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6T136W Too many DD names for the
 HLQNAME '*hlq_name*'. Only the
 first five values will be used**
Explanation:

The CPARM parameter file contain too many values for the parameter that is listed in the message.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6T137W DD names for the HLQNAME
 '*hlq_name*' are not specified**
Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6T138W DD name '*dd_name*' defined in the
 HLQDDDF Command is not found
 in the Source Job DD specification**
Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6T139W Invalid value was specified for
 DDNAME: DD name '*dd_name*'
 already exist**
Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6T140W Invalid value was specified for
 DDNAME: '*dd_name*' is reserved
 DD name**
Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

**CKZ6T141W Only one LISTDEF name
 supported. LISTDEFs with names
 other than '*listdef_name*' will be
 skipped**
Explanation:

Only one LISTDEF name is supported by the current version of Db2 Cloning Tool. The LD-LISTNAME parameter should be specified only once per profile.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T142W **Parameter LD-LISTNAME is not specified**

Explanation:

The LD-LISTNAME parameter is required if LISTDEFS are specified in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T143W **Incorrect {LD-LISTNAME | LD-ACTION | LD-TYPESPEC | LD-OBJTYPE}: 'incorrect_value'**

Explanation:

This message occurred due to incorrect input data in the CPARM file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T144W **Parameter {LD-ACTION | LD-TYPESPEC | LD-OBJTYPE | LD-FIRST-QUALIFIER} is not specified**

Explanation:

The parameter name that is listed in the message is required if LISTDEFS are specified in the CPARM file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T145W **Parameters LD-PARTLEVEL and LD-RI are mutually exclusive**

Explanation:

The LD-PARTLEVEL parameter has been specified and LD-RI is set to Y in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T146W **unsupported_qualifier is not supported for {TS | IS | TB | IX | DB | SG}**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T147W **Parameter LD-SECOND-QUALIFIER should be specified for the {TS | IS | TB | IX}**

Explanation:

If LISTDEFS are specified in the CPARM parameter file, the LD-SECOND-QUALIFIER parameter is required for the object type that is listed in the message

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T151W **Incorrect OM-TYPE: 'incorrect_value'. Value will be skipped**

Explanation:

The CPARM parameter file contains either unsupported or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T152W **The mismatch object type mismatch_object_type is already defined. New value of object mismatch return code will be skipped**

Explanation:

The mismatch object type that is listed in the message has already been defined. The new value will be ignored.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T156W **A different number of source and target values specified for the object_translate_type**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T161E **Current profile version is not supported**

Explanation:

Current versions of the load libraries do not work with new versions of profiles.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T162W **One of the values of ADVISORY-STATUS-VALUES is incorrect: 'incorrect_value'. Value will be skipped**

Explanation:

The CPARM parameter file contains either unsupported or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T163W **One of the values of RESTRICT-STATUS-VALUES is incorrect: 'incorrect_value'. Value will be skipped**

Explanation:

The CPARM parameter file contains either unsupported or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T164W **DM-PGM is equal to *program*. Therefore the *parameter_name* keyword was set to '*value*'**

Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T165W **Invalid data set name was entered for {JT-INPUT-DATASET | JT-OUTPUT-DATASET}**

Explanation:

This message occurred due to incorrect input data in the CPARM file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T166W **Invalid {SOURCE-SSID | TARGET-SSID}: '*ssid*'**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T167W **Invalid value was specified for {PROFILE-CREATED-TIMESTAMP | SOURCE-TOKEN | SOURCE-LOCATION | CATWORK-DSN-MASK | DATACLAS | MGMTCLAS | STORCLAS | LA-TO-LOGPOINT | LA-TO-TIMESTAMP}**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T168W **Reserved DD name was specified for '*parameter_name*'**

Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T169W ***parameter_name* is equal to 'Y'. Therefore the other *parameter_name* keyword was set to '*value*'**

Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T171W **The parameter '*job_card_parameter_name*' is present, but not specified. The values of '*job_card_parameter_name*' will be empty**

Explanation:

The CPARM parameter file contains incorrect information.

User response:

If needed, change the value of the described field manually.

CKZ6T181W **A different number of list items in job templates specified**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T211W **Incorrect {DDL-ATTRIBUTE-NAME | DDL-APPLY-TO-TYPE}: 'value'**

Explanation:

The CPARM file contains either unsupported or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T212W **Parameter DDL-ATTRIBUTE-NAME is not specified**

Explanation:

If the DDL ATTRIBUTE CHANGE command is specified in the CPARM parameter file, the parameter name that is listed in the message is required.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T213W **Invalid value of parameter {DDL-APPLY-TO-TYPE | DDL-SOURCE-VALUE | DDL-TARGET-VALUE} for the {STOGROUP | BUFFERPOOL | GBPCACHE | LOG | PRIQTY | SECQTY | TRACKMOD | CLOSE | DATACAPTURE}**

Explanation:

The CPARM parameter file contains either unsupported or incorrect information.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T276W **The JTV-NAME is initialized by reserved job template variable name**

Explanation:

The CPARM parameter file contains either incorrect information or information from the old versions of profiles.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T277W **Incorrect value of parameter JTV-NAME**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T291W **Invalid value of parameter {SJ-USER-DDN | TJ-USER-DDN | RJ-USER-DDN | TSJ-USER-DDN | STSJ-USER-DDN}: 'dd_name'. This is reserved DD name**

Explanation:

This message occurred due to incorrect input data in the CPARM parameter file.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T296W **The module CKZ00991 for masking validation is not found**

Explanation:

The CKZ00991 module that is required for data masking validation was not found.

User response:

Ensure that the correct and latest versions of all required load libraries are used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T297W **Unknown error of masking validation routine**

Explanation:

An error occurred in the data masking validation module.

User response:

Ensure that the correct and latest versions of all required load libraries are used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ6T298W **Invalid value for the MD-MASKRULE specified: *system_message***

Explanation:

The CPARM parameter file contains incorrect information.

User response:

If you changed the CPARM file manually, ensure that it contains accurate information.

CKZ6T301I ***module_version_information***

Explanation:

This message displays version information about the module that is running.

User response:

No action is required.

CKZ6T302E External modules initialization error

Explanation:

Some of the required external modules were not initialized.

User response:

Check previous warnings for more information.

CKZ70801I Version mismatch between source and target TABLE | INDEX, source: *source_object_creator.source_object_name - source_object_version_from_Db2_catalog*, target: *target_object_creator.target_object_name - target_object_version_from_Db2_catalog*

Explanation:

This informational message indicates that the version numbers for the specified source and target objects are different.

User response:

No action is required.

CKZ70802I Version mismatch for source TABLE *table_creator.table_name* between page set (*object_version_number_from_page_set*) and Db2 catalog (*object_version_number_from_Db2_catalog*)

Explanation:

This informational message indicates that the highest data version in the page set differs from the version in the Db2 catalog for the specified table.

User response:

No action is required.

CKZ70803W No system pages were found for source TABLE *source_table_creator.source_table_name* paired with versioned target TABLE *target_table_creator.target_table_name*

Explanation:

This warning message indicates that the source page set does not contain system pages with a schema definition for the specified table. This is required because the target table has active versioning.

User response:

Run the REPAIR utility with keyword

INSERTVERSIONPAGES on the source table space that contains the specified table, or drop and re-create the target table. Once this is done, resubmit the job.

CKZ70804W Data version in page set for source TABLE *table_name.table_creator* is greater than version in Db2 catalog. Version in page set: *table_version_from_page_set*, version in Db2 catalog: *table_version_from_Db2_catalog*

Explanation:

Db2 Cloning Tool detected that the data version for the specified table in the page set is greater than the version in Db2 catalog. This indicates that there is an inconsistency between the source page set and the source Db2 catalog.

User response:

Run the REPAIR CATALOG utility on the source object and resubmit the job.

CKZ70805W Data version for source TABLE *source_table_creator.source_table_name* (*source_data_version_from_page_set*) does not match version in Db2 catalog (*source_object_version_from_Db2_catalog*), which is greater than version of target TABLE *target_table_creator.target_table_name* (*target_object_version_from_Db2_catalog*)

Explanation:

The target Db2 subsystem version is V10, and the specified combination of source and target version can be resolved only with REPAIR CATALOG, which is not available on the target Db2 subsystem.

User response:

Run the REORG utility on the source object, or drop and re-create the target object. Once this is done, resubmit the job.

CKZ70806I REPAIR | REBUILD is required to resolve versioning mismatch between source TABLE | INDEX *source_table_creator.source_table_name* and target TABLE | INDEX *target_table_creator.target_table_name* but automatic REPAIR | REBUILD is disabled

Explanation:

The specified utility name is required for resolving versioning mismatch between the specified objects, but the cloning of these objects does not allow the

utility to run automatically. This message was not escalated to a warning message because the TB_VERSION mismatch return code was set to 0.

User response

Do one of the following:

- Enable automatic invocation of the specified utility by specifying either SET command parameter TARGET-JOB-REPAIR-EXECUTE(Y) or REBUILD-INDEXES-EXECUTE(Y), then resubmit the job.
- Specify TARGET-JOB-REPAIR-DDN to generate a REPAIR job, then run the REPAIR job.
- Add CKZINTRB DD to the target job JCL or specify the SET command parameter TARGET-JOB-INDEX-REBUILD-DDN to generate a REBUILD INDEX job, then run the REBUILD INDEX job.
- Run the specified utility manually after the Db2 Cloning Tool Table Space Cloning target job completes.

CKZ70807W **REPAIR | REBUILD is required to resolve versioning mismatch between source TABLE | INDEX *source_table_creator.source_table_name* and target TABLE | INDEX *target_table_creator.target_table_name* but automatic REPAIR | REBUILD is disabled**

Explanation:

The specified utility name is required for resolving versioning mismatch between the specified objects, but the cloning of these objects does not allow the utility to run automatically.

User response

Do one of the following:

- Enable automatic invocation of the specified utility by specifying either SET command parameter TARGET-JOB-REPAIR-EXECUTE(Y) or REBUILD-INDEXES-EXECUTE(Y), then resubmit the job.
- Specify TARGET-JOB-REPAIR-DDN to generate a REPAIR job, then run the REPAIR job.
- Add CKZINTRB DD to the target job JCL or specify the SET command parameter TARGET-JOB-INDEX-REBUILD-DDN to generate a REBUILD INDEX job, then run the REBUILD INDEX job.
- Run the specified utility manually after the Db2 Cloning Tool Table Space Cloning target job completes.

CKZ70808W **LOG-APPLY processing for source TABLE SPACE *source_database_name.source_table_space_name* resulted in an**

unexpected data version number. Expected version: *expected_version*, version from page set: *version_from_page_set*.

Explanation:

Db2 Cloning Tool detected an unexpected data version in a table space page set after LOG-APPLY processing

User response:

Run the REPAIR CATALOG utility on the specified table space and rerun the Db2 Cloning Tool Table Space Cloning source and target jobs.

CKZ70809W **Cloning source TABLE *source_table_creator.source_table_name* to non-zero version target TABLE *target_table_creator.target_table_name* with PGM(SRCIMCPY) resulted in page set without system pages for specified table**

Explanation:

Target table *target_table_creator.target_table_name* has active versioning, but cloning with PGM(SRCIMCPY) resulted in a page set that has no system pages for the specified table and therefore no previous versions in the system pages. The target Db2 subsystem requires a schema definition for previous versions in system pages. This may cause problems with target table availability.

User response:

There are two possible ways to resolve this problem. The first is to drop and re-create the target table. The second is to use a LOG-APPLY END-POINT that corresponds to a state where source table *source_table_creator.source_table_name* will have version 0 system pages. If the source Db2 subsystem has Epic 37738 installed, you can use the REPAIR utility with keyword INSERTVERSIONPAGES to create missing system pages.

CKZ70810W **REPAIR is required for target TABLE *table_creator.table_name*, but automatic REPAIR is disabled.**

Explanation:

Db2 Cloning Tool detected that REPAIR is required for the specified target table, but the cloning options do not allow the utility to run automatically.

User response

Do one of the following:

- Enable automatic invocation of the REPAIR utility by specifying SET command parameter TARGET-JOB-REPAIR-EXECUTE(Y) and resubmit the job.
- Specify TARGET-JOB-REPAIR-DDN to generate a REPAIR utility job, then run the REPAIR job.

- Run the REPAIR utility manually after Db2 Cloning Tool Table Space Cloning target job completes.

CKZ70811E REPAIR | REBUILD is required to resolve versioning mismatch between source TABLE | INDEX *source_table_creator.source_table_name* and target TABLE | INDEX *target_table_creator.target_table_name* but automatic REPAIR | REBUILD is disabled

Explanation:

The specified utility name is required for resolving versioning mismatch between the specified objects, but the cloning options do not allow the utility to run automatically.

User response

Do one of the following:

- Enable automatic invocation of the specified utility by specifying either SET command parameter TARGET-JOB-REPAIR-EXECUTE(Y) or REBUILD-INDEXES-EXECUTE(Y), then resubmit the job.
- Change the mismatch return code for TB_VERSION mismatch and resubmit the job.
- Specify TARGET-JOB-REPAIR-DDN to generate a REPAIR job, then run the REPAIR job.
- Add CKZINTRB DD to the target job JCL or specify the SET command parameter TARGET-JOB-INDEX-REBUILD-DDN to generate a REBUILD INDEX job, then run the REBUILD INDEX job.
- Run the specified utility manually after the Db2 Cloning Tool Table Space Cloning target job completes.

CKZ70812I No system pages were found for source TABLE *tb_creator.tb_name* paired with versioned target TABLE *tb_creator.tb_name*. May be fixed by re-creating, will be checked on re-create compatibility later

Explanation:

This message indicates that the source page set does not contain system pages with a schema definition for the specified table, and the target page set has active versioning. The target object will be re-created to resolve this issue.

User response:

No action is required.

CKZ70901I Subtask *subtask_number*, Consistent FLASHCOPY was found for *DB2_data_set_name* in *consistent_FlashCopy_data_set_na*

me PIT_RBA:
pit_rba_of_consistent_FlashCopy

Explanation:

A consistent FlashCopy for the specified data set was found and will be used as the copy source.

User response:

No action is required.

CKZ70902E Subtask *subtask_number*, CKZ00709 internal error. RC=*return_code* RS=*reason_code* INFO=*information_string*

Explanation:

An internal error occurred during USE-LAST-CONSISTENT-FLASHCOPY processing.

User response:

Contact IBM Software Support.

CKZ70903I Subtask *subtask_number*, IDCAMS control cards will follow

Explanation:

The next set of CKZ70904I messages will contain control cards that are passed to the IDCAMS utility.

User response:

No action is required.

CKZ70904I Subtask *subtask_number*, control_card_contents

Explanation:

This message contains an IDCAMS utility control card.

User response:

No action is required.

CKZ70905I Subtask *subtask_number*, IDCAMS output records will follow

Explanation:

The next set of CKZ70906I messages will output from the IDCAMS utility.

User response:

No action is required.

CKZ70906I Subtask *subtask_number*, IDCAMS_output_record

Explanation:

This message contains an IDCAMS output record.

User response:

No action is required.

CKZ70907E Subtask *subtask_number*, IDCAMS has returned non-zero RC: *return_code*

Explanation:

The IDCAMS utility returned a non-zero return code that is listed in the message.

User response:

Refer to the CKZ70906I messages to determine the reason of IDCAMS failure. If unable to determine or correct the problem, contact IBM Software Support.

CKZ70908I Subtask *subtask_number*, Data component name does not conform to DB2 naming conventions. IDCAMS ALTER will be called to fix it. Cluster: *cluster*
Data: *data*

Explanation:

Copying from consistent FLASHCOPY data sets may result in target DB2 data sets with data component names that do not correspond to DB2 naming conventions. This can lead to various DB2 errors. IDCAMS ALTER will be called to rename these data sets.

User response:

No action is required.

CKZ70909I Data set *data_set_name* no longer exists on source Db2, but found in FLASHCOPY data set *flashcopy_data_set*

Explanation:

The last consistent FlashCopy contains an extension data set that is no longer present on the source Db2. It will be included in the copy.

User response:

No action is required.

CKZ71000I Source subsystem *DB2_subsystem* is not a data sharing member

Explanation:

This message is informational.

User response:

No action is required.

CKZ71001I Source subsystem *DB2_subsystem* is a member of data sharing group *group_name*

Explanation:

This message is informational.

User response:

No action is required.

CKZ71002I Data sharing member=*data_sharing_member_name*, ID=*data_sharing_ID*, SSID=*DB2_subsystem*, CMDPRFX=*command_prefix*, STATUS=*status_of_this_member*, DB2V=*DB2_version_of_this_member*, SYS=*z/OS_system_name*

Explanation:

This message is informational.

User response:

No action is required.

CKZ71003I System checkpoint taken for subsystem *DB2_subsystem*, system *z/OS_subsystem_name*

Explanation:

This message is informational.

User response:

No action is required.

CKZ71005E Unable to find a checkpoint record

Explanation:

The checkpoint record could not be found in the BSDS.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ71006E Checkpoint record format error

Explanation:

The checkpoint record is in error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ71007I RBA / LRSN used for log apply is *value_in_hex ddname*

Explanation:

This message is informational. *ddname* is for the BSDS used.

User response:

No action is required.

CKZ71008I BSDS data sharing member=*data_sharing_member_name*, ID=*data_sharing_ID*, group=*group_name*, BSDS01=*BSDS01_data_set_for_this_ID*, BSDS02=*BSDS02_data_set_for_this_ID*

Explanation:

This message is informational. *ddname* is for the BSDS used.

User response:

No action is required.

CKZ71009E Unable to find any data sharing member records in the BSDS

Explanation:

The BSDS is in error. The BSDS data sets may be specified incorrectly.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ71010E **No DATA-SHARING-MEMBERS command found for member ID=*data_sharing_ID***

Explanation:
Unable to find all the information about this member.

User response:
Supply the DATA-SHARING-MEMBERS command and rerun the job.

CKZ71011E **Unable to find root record in the BSDS.**

Explanation:
This message indicates an error processing the BSDS data set.

User response:
If unable to resolve this error, contact IBM Software Support.

CKZ71012I **Using *RBA/LRSN_length* byte *RBA/LRSN*, flag=*DSNJCNVT flag_in_hex*.**

Explanation:
This message is informational. The RBA/LRSN length (6 or 10 bytes) used by log apply is provided in the message.

User response:
No action is required.

CKZ71014I ***subsystem_id*, CKPT *checkpoint1 checkpoint2 checkpoint3***

Explanation:
This message is informational. *subsystem_id* might be empty. This message provides checkpoint values that are retrieved after SET LOGLOAD command execution.

User response:
No action is required.

CKZ71101I **Subtask *subtask_number*, QUIESCE output open OK for DDname *ddname***

Explanation:
This message is informational.

User response:
No action is required.

CKZ71102I **Subtask *subtask_number*, QUIESCE commands to follow ...**

Explanation:
This message is informational.

User response:
No action is required.

CKZ71103I **Subtask *subtask_number*, *message_text***

Explanation:
This message is informational.

User response:
No action is required.

CKZ71104E **Subtask *subtask_number*, DSNUTILB attach error, RC=*return_code*, RS=*reason_code***

Explanation:
Unable to start DSNUTILB.

User response:
Contact IBM Software Support if unable to resolve this error.

CKZ71105I **Subtask *subtask_number*, table spaces quiesced at RBA/LRSN = *quiesce_point***

Explanation:
This message is informational.

User response:
No action is required.

CKZ71106I **Subtask *subtask_number*, clone table spaces quiesced at RBA/LRSN = *quiesce_point***

Explanation:
This message is informational.

User response:
No action is required.

CKZ71107E **Subtask *subtask_number*, DSNUTILB execute error, RC=*return_code*, RS=*reason_code*, text to follow...**

Explanation:
An error occurred during DSNUTILB execution.

User response:
Contact IBM Software Support if unable to resolve this error.

CKZ71108I **Subtask *subtask_number*, QUIESCE input open OK**

Explanation:
This message is informational.

User response:
No action is required.

CKZ71109I **Subtask *subtask_number*, QUIESCE messages to follow**

Explanation:
This message is informational.

User response:

No action is required.

CKZ71110I Subtask *subtask_number*,
text_of_QUIESCE_messages

Explanation:

This message is informational.

User response:

No action is required.

CKZ71111E Subtask *subtask_number*, Unable
to determine quiesce point

Explanation:

This message is an error.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ71301I Subtask *subtask_number*, REPAIR
output open OK for DD-NAME
ddname

Explanation:

This message is informational.

User response:

No action is required.

CKZ71302I Subtask *subtask_number*, REPAIR
commands to follow...

Explanation:

This message is informational.

User response:

No action is required.

CKZ71303I Subtask *subtask_number*,
message_text

Explanation:

This message is informational.

User response:

No action is required.

CKZ71304E Subtask *subtask_number*,
DSNUTILB attach error,
RC=*return_code*, RS=*reason_code*

Explanation:

Unable to start DSNUTILB.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ71307E Subtask *subtask_number*,
DSNUTILB execute error,
RC=*return_code*, RS=*reason_code*
text to follow...

Explanation:

An error occurred during DSNUTILB execution.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ71308I Subtask *subtask_number*, REPAIR
input open OK

Explanation:

This message is informational.

User response:

No action is required.

CKZ71309I Subtask *subtask_number*, REPAIR
messages to follow

Explanation:

This message is informational.

User response:

No action is required.

CKZ71310I Subtask *subtask_number*,
message_text

Explanation:

This message is informational.

User response:

No action is required.

CKZ71311I Utility *utility_name*, Maximum
RC=*return_code*

Explanation:

This message is informational.

User response:

No action is required.

CKZ71312W Utility *utility_name*, Maximum
return code=*return_code*

Explanation:

This message is a warning.

User response:

Determine the cause from the preceding utility messages. Correct the problem and resubmit the job.

CKZ71313E Utility *utility_name*, Maximum
return code=*return_code*

Explanation:

This message is an error.

User response:

Determine the cause from the preceding utility messages. Correct the problem and resubmit the job.

CKZ71400E Log apply requested data set
processing; however, no data set
by that name was found,
DSN=*dsname*

Explanation:

The data set listed in the message was not found.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ71401E **Log apply error return,
return_code, exit_name, dsname**

Explanation:

This message is an error.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ71402I **PGM(SRCIMCPY) specified, copy
has started**

Explanation:

This message is informational.

User response:

No action is required.

CKZ71403E **Subtask *subtask_number*,
PGM(*program*) specified, open for
the data set already processed**

Explanation:

This message is an error.

User response:

Contact IBM Software Support.

CKZ71404E **Subtask *subtask_number*,
PGM(*program*) specified, close for
the data set already processed**

Explanation:

This message is an error.

User response:

Contact IBM Software Support.

CKZ71405I **PGM(SRCVSCPY) specified, copy
has started**

Explanation:

This message is informational.

User response:

No action is required.

CKZ71406E **Log Apply init call failed,
RC=*return_code*, RS=*reason_code***

Explanation:

A severe error occurred when log apply was processing a source server request.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ71407W **Log apply init call completed with
warning(s) RC=*return_code*,
RS=*reason_code***

Explanation:

An error occurred when log apply was processing a source server request. It is possible that the error was generated from a single data set.

User response:

Check the messages that precede this message to determine which objects have an associated warning. The return code field in the source report cannot be used to determine which objects have a warning. Contact IBM Software Support if unable to resolve this error.

CKZ71408W **Log apply initialization warning,
RC=*return_code***

Explanation:

Log apply has initialized with one or more warnings.

User response:

Check the messages that precede this error message; refer to CKZ00031I for message output. Contact IBM Software Support if unable to resolve this error.

CKZ71409I **Encrypted target data set *dsname*
is reallocated as non-encrypted**

Explanation:

SRCIMCPY or SRCVSCPY had reallocated the data set that is listed in the message as non-encrypted. The data set was encrypted before the copy started. This condition is not treated as warning because ENCRYPTION-MISMATCH-RC(0).

User response:

No action is required.

CKZ71410W **Encrypted target data set *dsname*
is reallocated as non-encrypted**

Explanation:

SRCIMCPY or SRCVSCPY had reallocated the data set that is listed in the message as non-encrypted. The data set was encrypted before the copy started.

User response:

It is possible that this condition is caused by SMS rules. If this result is not expected, contact your systems programmer. If unable to resolve the issue, contact IBM Software Support.

CKZ71499I **Subtask *subtask_number*,
log_apply_call_type, value**

Explanation:

This message is informational. *value* can be empty or contain a data set name.

User response:

No action is required.

CKZ71501I **Subtask *subtask_number*,
REBUILD output open OK for DD-
NAME ddname**

Explanation:

This message is informational.

User response:
No action is required.

CKZ71502I Subtask *subtask_number*,
REBUILD commands to follow...

Explanation:
This message is informational.

User response:
No action is required.

CKZ71503I Subtask *subtask_number*,
message_text

Explanation:
This message is informational.

User response:
No action is required.

CKZ71504E Subtask *subtask_number*,
DSNUTILB attach error,
RC=*return_code*, RS=*reason_code*

Explanation:
Unable to start DSNUTILB.

User response:
If unable to resolve this error, contact IBM Software Support.

CKZ71505W There are some objects for
REBUILD but REBUILD-INDEXXES-
EXECUTE is disabled and
CKZINTRB DD is not specified

Explanation:
Db2 Cloning Tool detected objects that need REBUILD,
but options for automatic REBUILD are disabled.

User response:
Rerun the cloning process with automatic REBUILD
features enabled, or manually run the REBUILD utility
for the target objects.

CKZ71507E Subtask *subtask_number*,
DSNUTILB execute error,
RC=*return_code*, RS=*reason_code*
Text to follow...

Explanation:
An error occurred during DSNUTILB execution.

User response:
If unable to resolve this error, contact IBM Software Support.

CKZ71508I Subtask *subtask_number*,
REBUILD input open OK

Explanation:
This message is informational.

User response:

No action is required.

CKZ71509I Subtask *subtask_number*,
REBUILD messages to follow

Explanation:
This message is informational.

User response:
No action is required.

CKZ71510I Subtask *subtask_number*,
message_text

Explanation:
This message is informational.

User response:
No action is required.

CKZ71511I Utility *utility_name*, Maximum
return code=*return_code*

Explanation:
This message is informational.

User response:
No action is required.

CKZ71512W Utility *utility_name*, Maximum
return code=*return_code*

Explanation:
This message is a warning. The utility that issued the
warning is listed in the message, along with a return
code.

User response:
Determine the cause from the preceding utility
messages. Correct the problem and resubmit the job.

CKZ71513E Utility *utility_name*, Maximum
return code=*return_code*

Explanation:
This message is an error. The utility that issued the
error is listed in the message, along with a return code.

User response:
Determine the cause from the preceding utility
messages. Correct the problem and resubmit the job.

CKZ71514E Utility *utility_name*, *message_text*

Explanation:
This message is an error. The utility that issued the
error is listed in the message, along with error
message text.

User response:
Determine the cause from the utility messages.
Correct the problem and resubmit the job.

CKZ71600I Subtask *integer*,
utility_sprint_detail

Explanation:

This message is used to report the SYSPRINT output received from an invocation of the DSNUTILU stored procedure.

User response:

No action is required.

CKZ71601E Subtask *integer*, SYSPUNCH data set *syspunch_data_set_name* content exceeds the capacity of the utility control statement buffer

Explanation:

During an unload-load operation, the amount of SYSPUNCH control statements generated by the UNLOAD utility exceeds the size of the utility control statement buffer used by the DSNUTILU stored procedure. This may occur when a large number of LOAD control statement field specifications are generated by the UNLOAD utility.

User response:

Contact IBM Software Support. This message contains the name of the SYSPUNCH data set that caused this message to be issued. You can modify the SYSPUNCH data set content and run a LOAD utility to complete the cloning process for the object.

CKZ71602I Begin UNLOAD Status Report

Explanation

This message precedes a report that provides information about the data sets processed by UNLOAD/LOAD. The following columns are provided in the report:

SOURCE DATASET

The name of the source data set.

TARGET DATASET

The name of the target data set.

UNLOAD RC

The 2-digit return code from UNLOAD data set processing.

LOAD RC

The 2-digit return code from LOAD data set processing.

SPACE TYPE

The space type: IS (index space), TS (table space), LS (LOB table space), XS (XML table space)

CLONE

Y is a cloned space; B is a base space; blank if not cloned.

OBJ XLATE

One of the following:

- DB database name translation result
- TS table space name translation result
- IS index space name translation result

- IXC index creator translation result
- IXN index name translation result
- blank if no translation
- - if translation matches, but is not used
- + if translation matches and is used

User response:

No action is required.

CKZ71603I End UNLOAD Status Report

Explanation:

This message is informational and indicates the end of the report.

User response:

No action is required.

CKZ71604I Begin UNLOAD SIMULATE Status Report

Explanation

This message precedes a report that provides information about the data sets processed by UNLOAD in simulate mode. The following columns are provided in the report:

SOURCE DATASET

The name of the source data set.

TARGET DATASET

The name of the target data set.

UNLOAD RC

The 2-digit return code from UNLOAD data set processing.

LOAD RC

The 2-digit return code from LOAD data set processing.

SPACE TYPE

The space type: IS (index space), TS (table space), LS (LOB table space), XS (XML table space)

CLONE

Y is a cloned space; B is a base space; blank if not cloned.

OBJ XLATE

One of the following:

- DB database name translation result
- TS table space name translation result
- IS index space name translation result
- IXC index creator translation result
- IXN index name translation result
- blank if no translation
- - if translation matches, but is not used
- + if translation matches and is used

User response:

No action is required.

CKZ71605I **End UNLOAD SIMULATE Status Report**

Explanation:

This message is informational and indicates the end of the report.

User response:

No action is required.

CKZ71606I **Subtask *subtask_number*, calling LOAD utility with control statements from *data_set_name***

Explanation:

This message is informational.

User response:

No action is required.

CKZ71607I **Start of SYSPUNCH/SYSREC list, the following SYSPUNCH/SYSREC DSN should be available on target**

Explanation:

This message is informational. When using PGM(NONE), make sure that data sets from this list will be available on the target.

User response:

No action is required.

CKZ71608I **End of SYSPUNCH/SYSREC list**

Explanation:

This message is informational.

User response:

No action is required.

CKZ71609I **Begin LOAD Status Report**

Explanation:

This message is informational.

User response:

No action is required.

CKZ71610I **End LOAD Status Report**

Explanation:

This message is informational.

User response:

No action is required.

CKZ71611E **Subtask *subtask_number*, SYSPUNCH data set *data_set_name* is not found**

Explanation:

The SYSPUNCH data set that was read from SYNCDB2 is unavailable for target job.

User response:

Make sure that data set is available for the target job. Contact IBM Software Support if unable to resolve this error.

CKZ71612E **Subtask *subtask_number*, Unable to find valid *template_name* TEMPLATE control statement in data set specified in *ddname* DD**

Explanation:

The data set that is specified in the *ddname* DD does not contain a valid TEMPLATE control statement.

User response:

Make sure that the data set that is specified in *ddname* contains a valid TEMPLATE control statement.

CKZ71613E **Subtask *subtask_number*, *template_name* TEMPLATE control statement in data set specified in *ddname* DD exceeds the capacity of the utility control statement buffer**

Explanation:

The TEMPLATE control statement in the data set that is specified in the *ddname* DD exceeds the capacity of the utility control statement buffer.

User response:

Reduce the size of the TEMPLATE control statement.

CKZ71614I **Subtask *subtask_number*, utility_input_statements**

Explanation:

This message lists the control statements that were passed to the DSNUTILU stored procedure.

User response:

No action is required.

CKZ71615I **Subtask *subtask_number*, *procedure_name* will be used for operation**

Explanation:

Db2 Cloning Tool will use the *procedure_name* stored procedure to perform operations during UNLOAD/LOAD.

User response:

No action is required.

CKZ71616W **Subtask *subtask_number*, data set specified in DD *ddname* is empty**

Explanation:

The data set that is specified in DD *ddname* cannot be empty. It must contain a valid TEMPLATE.

User response:

Specify a valid TEMPLATE in the data set that is specified in DD *ddname* and resubmit the job.

| | | |
|-----------------------|---|------------------------|
| CKZ71700E | Subtask <i>subtask_number</i>, DSNUTILB execute Error, RC=<i>return_code</i>, RS=<i>reason_code</i>, text to follow... | No action is required. |
| Explanation: | An error occurred during DSNUTILB execution. | |
| User response: | If unable to resolve this error, contact IBM Software Support. | |
| CKZ71709I | Subtask <i>subtask_number</i>, DIAGNOSE messages to follow | |
| Explanation: | This message is informational. | |
| User response: | No action is required. | |
| CKZ71710I | Subtask <i>subtask_number</i>, <i>message_text</i> | |
| Explanation: | This message is informational. | |
| User response: | No action is required. | |
| CKZ71711E | Subtask <i>subtask_number</i>, unable to determine log point | |
| Explanation: | This message is an error. | |
| User response: | If unable to resolve this error, contact IBM Software Support. | |
| CKZ71702I | Subtask <i>subtask_number</i>, DIAGNOSE commands to follow... | |
| Explanation: | This message is informational. | |
| User response: | No action is required. | |
| CKZ71703I | Subtask <i>subtask_number</i>, <i>message_text</i> | |
| Explanation: | This message is informational. | |
| User response: | No action is required. | |
| CKZ71704E | Subtask <i>subtask_number</i>, DSNUTILB attach error, RC=<i>return_code</i>, RS=<i>reason_code</i> | |
| Explanation: | Unable to start DSNUTILB. | |
| User response: | If unable to resolve this error, contact IBM Software Support. | |
| CKZ71705I | Subtask <i>subtask_number</i>, subsystem log point = <i>hex_value</i> | |
| Explanation: | This message is informational. | |
| User response: | No action is required. | |
| CKZ71708I | Subtask <i>subtask_number</i>, DIAGNOSE input open OK | |
| Explanation: | This message is informational. | |
| User response: | | |
| CKZ71712E | DIAGNOSE command has failed | |
| Explanation: | This message is an error. | |
| User response: | Contact IBM Software Support, unless this error is accompanied by message CKZ71713E. | |
| CKZ71713E | Subtask <i>subtask_number</i>, APAR PI43759 not available | |
| Explanation: | This message is an error. | |
| User response: | Apply the Db2 APAR and rerun the job. | |
| CKZ72000I | Subtask <i>subtask_number</i>, <i>message</i> | |
| Explanation: | This message is informational and contains a message that is returned from the real-time statistics processing module. | |
| User response: | No action is required. | |
| CKZ72001W | Subtask <i>subtask_number</i>, <i>message</i> | |
| Explanation: | This message is a warning and contains a message that is returned from the real-time statistics processing module. | |
| User response: | | |

If unable to determine the cause of this warning, contact IBM Software Support.

CKZ72002E Subtask *subtask_number*,
message

Explanation:

This message is an error and contains a message that is returned from the real-time statistics processing module.

User response:

See messages CKZ72001W and CKZ72000I to determine the possible cause of error. If unable to determine the cause of this error, contact IBM Software Support.

CKZ72005I Subtask *subtask_number*, RTS
processing module has ended with
RC=return_code

Explanation:

This message is informational and includes the return code from real-time statistics processing.

User response:

No action is required.

CKZ72006I Subtask *subtask_number*,
Preparing to read RTS for objects
in DSN *source_data_set_name* |
Applying RTS from
rts_data_set_name

Explanation:

This message is informational.

User response:

No action is required.

CKZ72007E Subtask *subtask_number*, RTS
processing internal error.
RC=return_code RS=reason_code
INFO=additional_information

Explanation:

This message is an internal error that was detected during real-time statistics copy processing.

User response:

Contact IBM Software Support.

CKZ72008I Subtask *subtask_number*, Calling
RTS processing module

Explanation:

This message is informational.

User response:

No action is required.

CKZ72009I Subtask *subtask_number*,
applying RTS from *data_set_name*

Explanation:

This message is informational.

User response:

No action is required.

CKZ72101I Subtask *subtask_number*, XML
DOCID value for source | target
table *table_creator.table_name* is
value

Explanation:

This informational message lists the maximum XML DOCID sequence value for the source or target table.

User response:

No response is required.

CKZ72102I Subtask *subtask_number*, XML
DOCID value for source | target
table *table_creator.table_name* is
NULL. Update is not needed.

Explanation:

The table that is listed in the message has a NULL XML DOCID sequence value. A portion of the update may be skipped.

User response:

No response is required.

CKZ72103E Subtask *subtask_number*, unable
to fetch XML DOCID sequence
value for source | target table
table_creator.table_name

Explanation:

Information about the XML DOCID sequence for the specified table could not be obtained.

User response:

Contact IBM Software Support.

CKZ72104I Subtask *subtask_number*, found
DS member *member_name*:
member_status

Explanation:

This informational message indicates that the member that is listed in the message is a member of data sharing group on the target, and displays the member's status.

User response:

No action is required.

CKZ72105I Subtask *subtask_number*, current
member: *member_name*

Explanation:

This informational message indicates that the target job is connected to the data sharing member that is listed in the message.

User response:

No action is required.

CKZ72106I Subtask *subtask_number*, XML DOCID update module ended with RC=*return_code*

Explanation:

The XML DOCID update module completed with the return code that is listed in the message.

User response:

No action is required.

CKZ72107I Subtask *subtask_number*, message *text*

Explanation:

This message displays information from the XML DOCID update module.

User response:

No action is required.

CKZ72108W Subtask *subtask_number*, message *text*

Explanation:

This warning message displays information from the XML DOCID update module.

User response:

Consult other messages to determine the cause for the warning message. If unable to resolve the problem, contact IBM Software Support.

CKZ72109E Subtask *subtask_number*, message *text*

Explanation:

This error message displays information from the XML DOCID update module.

User response:

Consult other messages to determine the cause for the error message. If unable to resolve the problem, contact IBM Software Support.

CKZ72110W Target DB2 is in data sharing, but UPDATE-DOCID-JCL-DSN is not specified.

Explanation:

The target DB2 is in a data sharing group. To condition the other data sharing members of this group, Db2 Cloning Tool might need to generate additional jobs to the data set that is specified by the UPDATE-DOCID-JCL-DSN parameter. However, this parameter was not specified.

User response:

Correct the input and resubmit the job.

CKZ72111I Subtask *subtask_number*, SIMULATE(A) is in effect, XML DOCID sequences are not updated. Control cards for XML DOCID update utility will follow

Explanation:

This informational message indicates that XML DOCID sequences will not be updated because this is a simulation run.

User response:

No action is required.

CKZ72112I JCL for updating XML DOCID sequences for member *member* will be generated to *data_set_name*.

Explanation:

This informational message indicates that the JCL for updating XML DOCID sequences on the target data sharing member *member_name* will be generated to the specified data set.

User response:

No action is required.

CKZ72113I JCL for updating XML DOCID sequences for member *member_name* will not be generated because *reason*.

Explanation:

This informational message indicates that the JCL for updating XML DOCID sequences on the target data sharing member *member_name* will not be generated due to the specified reason.

User response:

No action is required.

CKZ72200I DDL generation started

Explanation:

This message is informational.

User response:

No action is required.

CKZ72201I DDL generation completed successfully

Explanation:

This message is informational.

User response:

No action is required.

CKZ72202W DDL generation completed with warning(s), RC=*return_code*

Explanation:

The DDL generation job did not complete successfully.

User response:

See the job log for DDL generation output messages. If unable to resolve the problem, contact IBM Software Support.

CKZ72203E DDL generation completed with error(s), RC= *return_code*

Explanation:

The DDL generation job did not complete successfully.

User response:

See the job log for DDL generation output messages. If unable to resolve the problem, contact IBM Software Support.

CKZ72204E *ddname DD open error, RC= return_code*

Explanation:

Db2 Cloning Tool was unable to open the DD *ddname* successfully.

User response:

If unable to resolve the problem, contact IBM Software Support.

CKZ72205E *Character char cannot be used as SQL terminator*

Explanation:

The specified *char* cannot be used as an SQL terminator for the executed DDL.

User response:

If PROCESS-TYPE(X) is used, correct the input and resubmit the job. Otherwise, contact IBM Software Support.

CKZ72206E *No records in DDL input DD ddname*

Explanation:

There are no DDL statements in the input DDL DD *ddname*. This message can occur when PROCESS-TYPE(X) is specified.

User response:

Check if the specified DD contains DDL statements. If it does, but the message still occurs, contact IBM Software Support. Otherwise, correct the input and resubmit the job.

CKZ72208I *number DDL statements were processed*

Explanation:

This message is informational.

User response:

No action is required.

CKZ72209I *number SQL error(s) was(were) ignored due to IGNORE-CREATE-OBJECT-EXISTS(Y) parameter*

Explanation:

This message is informational. It is displayed only when IGNORE-CREATE-OBJECT-EXISTS(Y) is specified and shows how many -454, -601, and -768 errors were received during DDL execution.

User response:

No action is required.

CKZ72210W *TCP/IP DDL processing warning. See server log for details*

Explanation:

A problem was encountered during DDL processing on the target system.

User response:

If unable to resolve the problem, contact IBM Software Support.

CKZ72211E *TCP/IP DDL processing error. See server log for details*

Explanation:

A problem was encountered during DDL processing on the target system.

User response:

If unable to resolve the problem, contact IBM Software Support.

CKZ72212I *number SQL error(s) was(were) ignored due to IGNORE-DROP-OBJECT-DOES-NOT-EXIST(Y) parameter*

Explanation:

This message is informational. It displays when IGNORE-DROP-OBJECT-DOES-NOT-EXIST(Y) is specified and shows how many -204 errors were received during DDL execution.

User response:

No action is required.

CKZ72213I *number DDL statements were processed with SQLERROR = -454*

Explanation:

This message is informational. It is displayed only when IGNORE-CREATE-OBJECT-EXISTS(Y) is specified, and shows how many -454 errors were received during DDL execution.

User response:

No action is required.

CKZ72300I *LISTDEF processing started*

Explanation:

This message is informational.

User response:

No action is required.

CKZ72301I *LISTDEF processing completed successfully*

Explanation:

This message is informational.

User response:

No action is required.

CKZ72302W **LISTDEF processing completed with warning(s), RC=return_code**

Explanation:

LISTDEF processing did not complete successfully.

User response:

See the job log for LISTDEF output messages. If unable to resolve the problem, contact IBM Software Support.

CKZ72303E **LISTDEF processing completed with error(s), RC=return_code**

Explanation:

LISTDEF processing did not complete successfully.

User response:

See the job log for LISTDEF output messages. If unable to resolve the problem, contact IBM Software Support.

CKZ72304E **ddname DD open error, RC=return_code**

Explanation:

Db2 Cloning Tool was unable to open the DD *ddname*.

User response:

If unable to resolve the problem, contact IBM Software Support.

CKZ76002I **Subtask *subtask_number*, inconsistent bit found in page applied from log, DSN=*data_set_name*, page=*DB2_page_number_in_hex***

Explanation:

This message is informational and may or may not indicate a problem with page data.

User response:

Determine if the page data is correct. If not, rerun the source and target job to reprocess the data set.

CKZ76003I **One or more data sets in this log apply group have no log records, MLDsn is *name_of_minilog_data_set***

Explanation:

This message is informational.

User response:

No action is required.

CKZ76004E **Subtask *subtask_number*, pending page already read and not processed, GrpPendPage=*hex_value_of_group_pending_page*, PGNUM=*hex_value_of_current_page*.**

Explanation:

An error has occurred during log apply processing.

User response:

If unable to resolve the error, contact IBM Software Support.

CKZ76005E **Subtask *subtask_number*, Pending space already read and not processed, GrpPendSPID=*pending_spid*, SpID=*spid***

Explanation:

An error occurred during log apply processing.

User response:

If unable to resolve the error, contact IBM Software Support.

CKZ76006E **Subtask *subtask_number*, bad eyecatcher, C call, *first_8_chars_of_eyecatcher***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ76009I **Subtask *subtask_number*, DSN(*data_set_name*), strings found =*number_of_XML_nodes_found_in_hex*, strings changed=*number_of_XML_string IDs_changed_in_hex***

Explanation:

This message provides information about XML processing.

User response:

No action is required.

CKZ76041W **Subtask *subtask_number*, no log page returned for space ID=*id*, page number=*page_num***

Explanation:

The log apply process did not find any logs to apply to the space.

User response:

No action is required. If you suspect a program error, contact IBM Software Support.

CKZ76088I **Subtask *subtask_number*, CKZG, correct=*expected_value*, error=*obtained_value*.**

Explanation:

This message accompanies message CKZ00012E and is used for diagnostic purposes.

User response:

Contact IBM Software Support.

CKZ76092I Subtask *subtask_number*, log apply *call_type* call successful

Explanation:

This message is informational.

User response:

No action is required.

CKZ76093W Subtask *subtask_number*, log apply *call_type* call warning, RC=*return_code*

Explanation:

This is a warning. Normally the return code is 4.

User response:

Contact IBM Software Support if unable to resolve this error.

CKZ76094E Subtask *subtask_number*, log apply *call_type* call failure, RC=*return_code*

Explanation:

This is an error. Normally the return code is 8 or 12.

User response:

Ensure that the minilog data set can be allocated if this is an OPEN call type. Contact IBM Software Support if unable to resolve this error.

CKZ76098I Subtask *subtask_number*, CKZG-C, *message_text*

Explanation:

A message from a page call. It could be informational, a warning or an error.

User response:

No action is required, unless the call ends with an error.

CKZ76601I Subtask *subtask_number*, Calling CKZ00995 for *data_set_name*

Explanation:

This informational message indicates the start of page processing for the data set that is listed in the message.

User response:

No action is required.

CKZ76602I Subtask *subtask_number*, CKZ00995 returned *return_code*

Explanation:

This informational message indicates that the page processing module ended with RC=*return_code*.

User response:

No action is required.

CKZ76603I Subtask *subtask_number*, *message_text*

Explanation:

This message is followed by informational message text from the page processing module.

User response:

No action is required. Refer to the *message_text* for additional information.

CKZ76604W Subtask *subtask_number*, *message_text*

Explanation:

This message is followed by warning message text from the page processing module.

User response:

Refer to the *message_text* for additional guidance.

CKZ76605E Subtask *subtask_number*, *message_text*

Explanation:

This message is followed by error message text from the page processing module.

User response:

Refer to the *message_text* for additional guidance.

CKZ76805E SUBTASK nn, SQL INSERT STATEMENT TO ADD STRING ssss TO CATALOG, ERROR, RC=rrr

Explanation:

This is a probable DB2 error. nn = subtask number ssss = first 80 characters of the string rrr = return code

User response:

Correct the error and resubmit the job.

CKZ76806I SUBTASK nn, SQL INSERT STATEMENTS FOR XML STRING IDS: ddd SUCCESSFUL

Explanation:

This message is informational. nn = subtask number ddd = decimal number

User response:

No action is required.

CKZ76807E SUBTASK nn, SQL INSERT STATEMENTS FOR XML STRING IDS: ddd SUCCESSFUL, eee UNSUCCESSFUL

Explanation:

This is a summary message. nn = subtask number ddd = decimal number of successful INSERTs eee = decimal number of unsuccessful INSERTs

User response:

See other error message(s) for the cause.

CKZ76810I Subtask *subtask_number*, target string ID created, source ID=*source_string_number_in_hex*,

target
ID=*target_string_id_now_in_catalog_in_hex*

Explanation:

This message is informational.

User response:

No action is required.

CKZ76811E **SUBTASK nn, TARGET STRING ID hhh NOT FOUND IN CATALOG, RC=rrr, RS=sss**

Explanation:

This is a possible DB2 error. The INSERT to the XML column completed normally, but Db2 Cloning Tool Table Space Cloning is unable to read the String ID from the catalog. nn = subtask number hhh = target hex string id rrr = return code sss = reason code

User response:

Determine if this is a DB2 error by querying the catalog. If not, contact IBM Software Support.

CKZ76901E **Subtask *subtask_number*, No TABLEDEF command found for table *table_creator.table_name***

Explanation:

This is an error message. The SYNCDB2 file may be corrupted.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ76902E **Subtask *subtask_number*, No MASKCMD and/or TABLEDEF command(s) available**

Explanation:

This is an error message. The SYNCDB2 file may be corrupted.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ76903E **Subtask *subtask_number*, Reason *reason_code*, cannot find COLNO for column *column_name*, table *table_creator.table_name***

Explanation:

This is an error message. The SYNCDB2 file may be corrupted.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ78003E **SUBTASK nn, BAD DATA MOVER VALUE**

Explanation:

This is an internal error. The input was validated during initialization, thus the value has been corrupted. nn = subtask number

User response:

Contact IBM Software Support.

CKZ78005E **SUBTASK nn, ADDRSSU HAS ENDED WITH A RETURN CODE > 8**

Explanation:

This message is an error. DSS is unable to process any more copy requests. nn = subtask number

User response:

Determine the problem using DSS messages and resubmit.

CKZ78006I **NO STARTS DONE FOR THE TARGET OBJECTS, TO BE DONE AT THE END OF TARGET JOB**

Explanation:

This message is informational. The target objects remain stopped until the target job has run.

User response:

No action is required.

CKZ78024I **SUBTASK nn, DDNAME(ddname), ALLOCATION OK**

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ78032W **SUBTASK nn, ERROR PARSING ttt; ONE OR MORE DSS DATA SET RETURN CODES MAY COME FROM DSS FINAL RETURN CODE**

Explanation:

While attempting to parse the DSS output, an error has occurred. This may prevent obtaining the return code for a particular data set. If so, the final DSS return code will be assigned to all those data sets without an explicit return code. nn = subtask number ttt = return value trying to parse

User response:

Check the individual return codes as printed out in CKZPRINT.

CKZ78041I **SUBTASK nn, DSS OUTPUT OPEN OK FOR DDNAME DDNAME**

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ78042I **SUBTASK nn, DSS COMMANDS TO FOLLOW ...**

Explanation:

This message is informational. See CKZ78043I for the command text. nn = subtask number

User response:

No action is required.

CKZ78043I SUBTASK nn, ccc

Explanation:

This message is informational. nn = subtask number
ccc = command line sent to DSS

User response:

No action is required.

CKZ78044I SUBTASK nn, DSS INPUT OPEN OK

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

**CKZ78045I SUBTASK *subtask_number*, DSS
MESSAGES TO FOLLOW**

Explanation:

This message is informational. See CKZ78043I for the message text.

User response:

No action is required.

**CKZ78047I SUBTASK nn, lll, COPY FROM ddd
TO eee, COMPLETED, RETURN
CODE IS ZERO**

Explanation:

This message is informational. nn = subtask number lll = copy program name ddd = source data set name eee = target data set name

User response:

No action is required.

**CKZ78048W SUBTASK nn, lll, COPY FROM ddd
TO eee, COMPLETED WITH
WARNINGS, RETURN CODE IS 4**

Explanation:

This message is a warning from DSS. If MAX_COPY_RC is 4 or lower, the job will terminate. nn = subtask number lll = copy program name ddd = source data set name eee = target data set name

User response:

Determine the cause of the error using DSS output messages and resubmit the job.

**CKZ78049E Subtask *subtask_number*,
copy_program_name, copy from
source_data_set TO
target_data_set, failed, return
code is *DSS_return_code***

Explanation:

This message is an error from DSS. If MAX_COPY_RC is 8, the job will continue. This data set did not copy correctly and must be copied again.

User response:

Determine the cause of the error using DSS output messages and resubmit the source job.

**CKZ78050E ADRDSSU has ended with a fatal
error.**

Explanation:

This message indicates an error has been found by ADRDSSU that is severe enough to stop copy processing.

User response:

If unable to determine the cause of the ADRDSSU error, contact IBM Software Support.

**CKZ78061W SUBTASK nn, UNABLE TO
CONNECT TO SSID *ssss* USING
CAF, CAN NOT ISSUE STOP
COMMANDS**

Explanation:

This is a warning. CKZ will attempt copies if MAX_RC=4. nn = subtask number *ssss* = DB2 subsystem

User response:

Contact your system programmer.

**CKZ78071W SUBTASK nn, UNABLE TO
CONNECT TO SSID *ssss* USING
CAF, CAN NOT ISSUE START
COMMANDS**

Explanation:

This is a warning. nn = subtask number *ssss* = DB2 subsystem

User response:

Contact your system programmer.

**CKZ78100I Template in DD *input_ddname1*
successfully processed and
written to DD *output_ddname2***

Explanation:

This message is informational.

User response:

No action is required.

**CKZ78101E Input template DD *ddname*,
unknown variable detected,
*variable_name***

Explanation:

The input DD has a variable name that is not recognized.

User response:

Correct the input variable name and resubmit.

CKZ78102I Maximum number of data sets per DSS call, *max_number_in_decimal*, number of data sets requested, *number_of_data_sets_to_copy_in_decimal*, number of DSS job steps required, *number_of_DSS_steps_required*.

Explanation:

This message is informational. It indicates that multiple steps must be coded in the input job template.

User response:

Add the number of steps to the input template(s).

CKZ78104E Input template DD *ddname*, illegal variable, *variable_name*

Explanation:

The input DD had a variable name that was illegal. A variable cannot have more than two ampersands (&). Both ampersands must be at the beginning of the variable name.

User response:

Correct the input variable name and resubmit.

CKZ78105E Input template DD *ddname*, variable value *variable_value* will not fit in 71 columns

Explanation:

The input DD has a variable name that is too long for the line.

User response:

Correct the input variable, or move the variable closer to column 1, and resubmit.

CKZ78106I No matching objects for input template *input_ddname1*, output template *output_ddname2*

Explanation:

This message is informational.

User response:

No action is required.

CKZ78107E Input template DD *ddname*, block variable *variable_name* is out of sequence

Explanation:

The input DD has a variable name that is not in the correct sequence. Variables that begin with &&BEGx (begin) and end with &&ENDx (end) must be in the correct sequence and cannot be nested.

User response:

Use the copy job template samples to determine the correct sequence. Correct the input variables and resubmit.

CKZ78108E Input template DD *ddname*, unrecognized template

Explanation:

The input DD has a template that cannot be processed.

User response:

Use the copy job template samples to determine the correct card specification. Correct the input variables and resubmit.

CKZ78109E Input template DD *ddname*, maximum number of *number_of_variables* variables exceeded

Explanation:

The input DD has specified more than 255 data sets in a DSS COPY statement.

User response:

Reduce the number of data sets to 255 or less, or use a data set list processing variable such as SRCDSNL or PAIRDSNL. Resubmit the job.

CKZ78110E Input template DD *ddname*, nest processing error, level *decimal_nest_level*

Explanation:

The input DD has specified an incorrect nest level for block variables.

User response:

Use the copy job template samples to determine the correct card specification. Correct the input variables and resubmit.

CKZ78111E Input template DD *ddname*, too many data set variables allowed *number_in_parmlib*, found *number_in_template*

Explanation:

The input DD has specified more than the DSNS_PER_COPY parameter in PARMLIB allows.

User response:

Reduce the number of data sets to the number that is specified in PARMLIB or use a list processing variable such as SRCDSNL or PAIRDSNL. Resubmit the job.

CKZ78112E Input template DD *ddname*, the number of *first_variable_name* variables, *num_first_variables*, must equal the number of *second_variable_name* variables, *num_second_variables*

Explanation:

The input DD has used &&SRCDSN0, &&SRCDSN1, &&TRGDSN0, or &&TRGDSN1 to create the copy template, and the number specified for the first

variable that is listed in the message does not match the number for the second variable.

User response:

Use the copy job template samples to determine the correct card specification. Correct the input variables and resubmit.

CKZ78113I **Input template DD *ddname*, the number of data sets per copy has changed from *num_data_sets_per_copy_in_parmlib* to *num_data_sets_per_copy_in_input_template*.**

Explanation:

This message is informational. The input DD used &&SRCDSNO, &&SRCDN1, &&TRGDSNO, or &&TRGDSN1 to create the copy template, and the number specified for data sets per COPY command is different than DSNS_PER_COPY parameter in PARMLIB. The value in PARMLIB is overridden by the new value.

User response:

No action is required.

CKZ78114I **Number of data sets available for job template processing: *TS=num_of_data_sets*, *IS=num_of_data_sets***

Explanation:

This message is informational.

User response:

No action is required.

CKZ78115E **The DD name specified for DSS commands in the DATA-MOVER command could not be found**

Explanation:

The input DD that is specified in the DATA-MOVER command cannot be located in the JCL.

User response:

Add the DD card for the input DD to the JCL and resubmit the job.

CKZ78116W **A JOB TEMPLATE WAS SPECIFIED, HOWEVER NO DATASETS MEET THE COPY CRITERIA**

Explanation:

This message is a warning. No job template was completed as no target data sets can be found.

User response:

Drop the JOB-TEMPLATE command or re-specify the objects to be processed so that one or more target data sets can be copied.

CKZ78117E **JOB CARD variable found, but no jobcards preceded it**

Explanation:

An incorrect template was passed to Db2 Cloning Tool.

User response:

Correct template and resubmit the job.

CKZ78118E **Input template DD *ddname*, object variable *object_variable_name* is incorrect.**

Explanation:

An error has occurred in the input job template.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ78119I **Number of *source / target* objects available for job template processing: *TS=decimal_number_of_table-space_objects*, *IS=decimal_number_of_index_objects*.**

Explanation:

This message is informational. It indicates which subtask requests have been purged due to an error.

User response:

No action is required.

CKZ78120E **Input template DD *ddname*, both data set and object variables are specified.**

Explanation:

An error occurred during job template input processing.

User response:

If unable to resolve the error, contact IBM Software Support.

CKZ78121E **Input template DD *ddname*, object variable *variable_name* was specified but no objects of that type were found**

Explanation:

An error occurred during job template input processing.

User response:

If unable to resolve the error, contact IBM Software Support.

CKZ78123E **Input Template DD *ddname*, Object Variable *variable* is Invalid in the Target Job**

Explanation:

This message indicates an error. *ddname* is the DD containing the invalid variable and *variable* is the invalid variable in the job template.

User response:

Correct the job template input variable and resubmit the job.

CKZ78124W **Subtask *subtask_number*, input template DDNAME references *job_type missing_object* but none were found.**

Explanation:

A template that is referenced by the job has no objects in the job.

User response:

Correct the input template and resubmit the job.

CKZ78201E **SUBTASK nn, mmm, RC=rrr, RS=sss, LOAD FAIL**

Explanation:

This message indicates an error. Module mmm was not loaded. nn = subtask number mmm = module that could not be loaded rrr = return code sss = reason code

User response:

This is an EMC module. Ensure the EMC modules are in the linklist or have been added to the source job steplib and resubmit the job.

CKZ78202I **EMCAPI VERSION=vvv, RELEASE=rrr, LEVEL=lll, PTF=ppp**

Explanation:

This message is informational. vvv = version number rrr = release number lll = level number ppp = PTF number

User response:

No action is required.

CKZ78203E **SUBTASK *subtask_number*, error_description, RC=return_code, RS=reason_code**

Explanation:

This message indicates an error. A brief description of the error is contained in the message, as well as the return and reason codes. Because the EMCAPI call return code exceeded the value of MAX-COPY-RC, this message is considered an error.

User response:

Check all EMC warning and error messages to determine the correct action to take.

CKZ78204I **SUBTASK nn, ttt**

Explanation:

This message is informational. nn = subtask number ttt = text of the EMC message

User response:

No action is required.

CKZ78205I **SUBTASK nn, EMC MESSAGES TO FOLLOW**

Explanation:

This message is informational. nn = subtask number

User response:

No action is required.

CKZ78206E **SUBTASK nn, UNEXPECTED RETURN FROM EMC COPY, RC=rrr**

Explanation:

This message indicates an error. nn = subtask number rrr = returned code

User response:

Check all EMC messages. If unable to determine from those messages what the problem is, contact IBM Software Support.

CKZ78207E **SUBTASK nn, EMCAPI HAS ENDED WITH A RETURN CODE > 8**

Explanation:

This message indicates an error. nn = subtask number

User response:

Check all EMC messages. If unable to determine from those messages what the problem is, contact IBM Software Support.

CKZ78208W **SUBTASK nn, EMCAPI MESSAGE SYNC ERROR, RC=rrr, RS=sss**

Explanation:

This message is a warning. nn = subtask number rrr = return code sss = reason code

User response:

Contact IBM Software Support.

CKZ78209W **SUBTASK nn, EMCAPI MESSAGE, SUMMARY RC rrr, Does Not Match Highest Dataset RC ddd**

Explanation:

This message is a warning. nn = subtask number rrr = return code from ESNP440I sss = highest return code from ESNP471I messages

User response:

Contact IBM Software Support.

CKZ78210W **SUBTASK nn, ERROR PARSING sss; ONE OR MORE EMC DATASET RETURN CODES MAY COME FROM EMC FINAL RETURN CODE**

Explanation:

This message is a warning. Any copy return codes not processed correctly, will be set using ESNP440I. See

CKZ78212W for the names of the data sets affected.
nn = subtask number sss = string not found

User response:

Contact IBM Software Support.

**CKZ78211E SUBTASK nn, ERROR PARSING
 sss, CANNOT FIND THE FINAL
 RETURN CODE**

Explanation:

This message indicates an error. nn = subtask number
sss = string not found

User response:

Contact IBM Software Support.

**CKZ78212W SUBTASK nn, DATASET dsname,
 CANNOT FIND RC, RC COMES
 FROM THE FINAL RETURN
 MESSAGE, ESNP440I**

Explanation:

This message is a warning. The final return code for
this data set will be set using ESNP440I. nn = subtask
number sss = string not found

User response:

Contact IBM Software Support.

**CKZ78213W SUBTASK *subtask_number*,
 warning_description,
 RC=*return_code*, RS=*reason_code***

Explanation:

This message is a warning. A brief description of the
warning or failure is contained in the message, as well
as the return and reason codes. Because the EMC-API
call return code did not exceed the value of MAX-
COPY-RC, this message is considered a warning.

User response:

Check all EMC warning and error messages to
determine the correct action to take.

CKZ7D000E Control file API is not initialized

Explanation:

An internal error occurred during control file API
initialization.

User response:

Contact IBM Software Support. Have available the
listing that contains this message.

**CKZ7D001I Export of DB2 subsystems
 information started**

Explanation:

The message marks the beginning of export of DB2
subsystems information.

User response:

No action is required.

**CKZ7D002I Import of DB2 subsystems
 information started**

Explanation:

The message marks the beginning of import of DB2
subsystems information.

User response:

No action is required.

**CKZ7D003I Export of DB2 subsystem *ssid*
 completed successfully**

Explanation:

The message marks the successful completion of the
export of the DB2 subsystem that is listed in the
messages.

User response:

No action is required.

**CKZ7D004E The first keyword in the DB2
 systems parameter file must be
 SSID**

Explanation:

An error occurred due to incorrect input data in the
SPARM file.

User response:

Ensure that the correct SPARM file was specified. If
you changed the SPARM file manually, ensure that it
contains accurate information.

**CKZ7D005E An internal error occurred while
 reading DB2 control file:
 *system_message***

Explanation:

An error occurred reading the DB2 control file.

User response:

Ensure that the correct and latest versions of all
required load libraries are being used. If unable to
determine the cause of this error, contact IBM
Software Support. Have available the listing that
contains this message.

**CKZ7D006W The DB2 systems parameter file
 does not contain any parameters**

Explanation:

There is no information about the DB2 systems in the
SPARM parameter file.

User response:

Ensure that the correct SPARM file was selected.

**CKZ7D007E Export of DB2 subsystems
 information failed**

Explanation:

An internal error occurred.

User response:

Check previous messages for more information.

**CKZ7D008W SSID parameter has invalid value:
*incorrect_value***

Explanation:

An invalid value for the SSID parameter was specified in the SPARM file.

User response:

If you plan to import that SSID in the future, correct the value of the SSID parameter in the SPARM file. If you changed the SPARM file manually, ensure that it contains accurate information.

**CKZ7D009I Export of DB2 subsystems
information finished**

Explanation:

The export of the DB2 subsystem information completed successfully.

User response:

No action is required.

**CKZ7D010I Import of DB2 subsystem *ssid*
started**

Explanation:

The message marks the beginning of import of the DB2 subsystem that is listed in the message.

User response:

No action is required.

**CKZ7D011W The list of SSIDs for export is
empty**

Explanation:

Source and target DB2 SSIDs were not specified for all exported profiles.

User response:

Specify source and target DB2 SSIDs for all exported profiles in the ISPF UI and run the export utility again.

**CKZ7D012W The parameter *parameter_name* is
already defined. New value will be
skipped**

Explanation:

The parameter name that is listed in the message has already been defined. The new value will be ignored.

User response:

If you changed the SPARM file manually, ensure that it contains accurate information.

**CKZ7D013W Export of DB2 subsystem *ssid*
failed. DB2 subsystem was not
found in DB2 control file**

Explanation:

The DB2 subsystem that is listed in the message will not be exported because the subsystem was not found in the DB2 control file.

User response:

Ensure that the correct DB2 control file was used.

**CKZ7D014W The unsupported parameter will
be skipped: *unknown_parameter***

Explanation:

The SPARM file contain unknown parameters.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If you changed the SPARM file manually, ensure that it contains accurate information.

**CKZ7D015I The DB2 subsystem *ssid* already
exists**

Explanation:

The subsystem that is listed in the message already exists.

User response:

No action is required.

**CKZ7D016I Import of DB2 subsystem *ssid*
completed successfully**

Explanation:

The subsystem that is listed in the message has been imported.

User response:

No action is required.

**CKZ7D017I Import of DB2 subsystem *ssid*
completed successfully. DB2
subsystem was skipped**

Explanation:

The subsystem that is listed in the message was not imported because there is an existing subsystem with the same name.

User response:

No action is required.

**CKZ7D018E Import of DB2 subsystems
information failed**

Explanation:

An error occurred during import of DB2 subsystems.

User response:

Check previous messages for more information.

**CKZ7D019I Import of DB2 subsystems
information finished**

Explanation:

The message marks the completion of the import of the DB2 subsystems information.

User response:

No action is required.

| | | |
|-----------------------|---|--|
| CKZ7D020E | An internal error occurred while writing to the DB2 control file: <i>system_message</i> | maximum length. Value will be truncated |
| Explanation: | The utility could not write to the DB2 control file. The reason for the error is listed in the message. | |
| User response: | Ensure that the correct and latest versions of all required load libraries are being used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message. | |
| CKZ7D021W | Too many values for parameter SDSNEXIT DDF-ALIAS DDF-DYNAMIC-ALIAS. Value will be ignored: <i>excess_value</i> | |
| Explanation: | The SPARM file contains too many values for the parameter that is listed in the message. | |
| User response: | If you changed the SPARM file manually, make sure that it contains accurate information. | |
| CKZ7D022I | The value of SDSNEXIT parameter is saved as LoadLib1 LoadLib2 LoadLib3 LoadLib4 LoadLib5 | |
| Explanation: | The SDSNEXIT parameter in the SPARM file was specified. This message indicates one or more load libraries that will be used. | |
| User response: | No action is required. | |
| CKZ7D023W | Too many values for SDSNLOAD. The SDSNEXIT parameter will be ignored | |
| Explanation: | The SPARM file contains too many values for the parameters that are listed in the message. | |
| User response: | If you changed the SPARM file manually, ensure that it contains accurate information. | |
| CKZ7D024E | The data set is not a DB2 control file: DSN = '<i>data_set_name</i>' | |
| Explanation: | An invalid data set name was specified for the DB2 control file. | |
| User response: | Specify a valid DB2 control file and resubmit the job. | |
| CKZ7D027W | The length of parameter value <i>parameter_name</i> exceeds the | |
| | | maximum length. Value will be truncated |
| Explanation: | The parameter value that is listed in the message is too long. The value will be truncated to the allowed length. | |
| User response: | If you changed the SPARM file manually, ensure that it contains accurate information. | |
| CKZ7D028W | <i>parameter_name</i> parameter has incorrect value. Value will be changed to the empty value | |
| Explanation: | The parameter value that is listed in the message has an incorrect value. The value will be removed. | |
| User response: | If you changed the SPARM file manually, ensure that it contains accurate information. | |
| CKZ7D029E | An internal error occurred while working with DB2 control file: <i>system_message</i> | |
| Explanation: | An internal error occurred. | |
| User response: | Ensure that the correct and latest versions of all required load libraries are being used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message. | |
| CKZ7D030W | Parameter {SPECIAL-DSNZPARM SYSVCAT} should be specified when SYSTEM-TYPE = T | |
| Explanation: | The SPARM parameter file does not contain enough information. | |
| User response: | If you changed the SPARM file manually, ensure that it contains accurate information. | |
| CKZ7D031W | SYSTEM-TYPE parameter value will be changed to a blank | |
| Explanation: | The SPARM parameter file does not contain enough information; therefore the value of the SYSTEM-TYPE parameter will be changed to blank. | |
| User response: | Check the previous message for more information. | |
| CKZ7D032W | Import of DB2 subsystem <i>ssid</i> completed with warnings | |
| Explanation: | | |

The subsystem that is listed in the message has been imported. Warnings were encountered during the import.

User response:

Check previous messages for more information.

CKZ7D033E Import of DB2 subsystem *ssid* failed. Invalid values

Explanation:

The subsystem that is listed in the message has not been imported because invalid values were encountered during the import.

User response:

Check previous warnings for more information.

CKZ7D034I Simulate mode is enabled. No changes will be added into the Db2 control file

Explanation:

This message is informational.

User response:

No action is required.

CKZ7D035W *parameter_name* parameter has an incorrect value. Value will be ignored

Explanation:

The parameter value that is listed in the message has an incorrect value. The value will be removed.

User response:

If you changed the SPARM file manually, ensure that it contains accurate information.

CKZ7D051E Parameter file data set does not exist

Explanation:

The SPARM parameter file is required.

User response:

Specify a valid path for the SPARM file and resubmit the job.

CKZ7D052E Parameter file data set has invalid type

Explanation:

The SPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Specify the SPARM file as a file with the correct data type and resubmit the job.

CKZ7D053E Failed to determine attributes of parameter file data set

Explanation:

Validation of SPARM file attributes failed. The SPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Ensure that the SPARM file with correct attributes is specified. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ7D054E Parameter file data set has invalid record format or record length

Explanation:

The SPARM parameter file must be defined as RECFM=FB, LRECL=80 and should be a member of a PDS or PDSE.

User response:

Specify the SPARM file as a file with correct attributes and resubmit the job.

CKZ7D055E An error occurred while reading the parameter file (line *n*): *system_message*

Explanation:

The utility could not read from the SPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ7D056E Invalid parameter file (line *n*): Incorrect parameter definition

Explanation:

An error occurred due to incorrect input data in the SPARM file.

User response:

Check the documentation for parameter file syntax and correct the syntax errors before resubmitting the job. If you changed the SPARM file manually, ensure that it contains accurate information.

CKZ7D057E An error occurred while writing to the parameter file: *system_message*

Explanation:

The utility could not write to the SPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ7D058E Cannot open parameter file: DSN = '*data_set_name*'[Member = '*member*']. *system_message*

Explanation:

The utility could not open the SPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ7D059E Cannot open parameter file: DDN = *dd_name*. *system_message*

Explanation:

The utility could not open the SPARM parameter file. The reason for the error is listed in the message.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ7D081E Unable to get current user ID

Explanation:

The utility could not get information about the current user ID.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ7D082E Unable to get current time

Explanation:

The utility could not get information about the current time.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ8B400I *module_version_information*

Explanation:

This message displays version information about the module that is running.

User response:

No action is required.

CKZ8B401I Beginning of the command file syntax checking | End of the command file syntax checking

Explanation:

This message marks either the beginning or the end of file syntax checking.

User response:

No action is required.

CKZ8B402I Initialization completed

Explanation:

This message indicates that the module completed initialization.

User response:

No action is required.

CKZ8B403E No commands were entered

Explanation:

The import or export utility was submitted, but there are no commands in the JCL.

User response:

Add commands to the import or export job and resubmit.

CKZ8B404I The number of commands for executions: *number_of_commands*

Explanation:

The number of commands that will be run is listed in the message.

User response:

No action is required.

CKZ8B405I EXPORT command | IMPORT command

Explanation:

This message indicates whether the command that is running is the EXPORT or the IMPORT command.

User response:

No action is required.

CKZ8B406I EXPORT command | IMPORT command | MIGRATE command *keyword = parameter_value*

Explanation

This message displays the command, a keyword, and the value for the keyword. Possible keywords are:

- PROFILE-TYPE
- PROFILE-REPOSITORY-HLQ
- CONTROL-FILE-DSN
- PARM-DSN
- CPARM-MEMBER
- SPARM-MEMBER
- CPARM-DDN
- SPARM-DDN
- PROFILE-LIKE
- CREATOR-LIKE
- IMPORT-ACTION-IF-PROFILE-EXISTS
- IMPORT-ACTION-IF-SUBSYSTEM-EXISTS
- USE-ORIGINAL-CREATORS
- SIMULATE

- SOURCE-REPOSITORY-HLQ
- TARGET-REPOSITORY-HLQ
- MIGRATE-ACTION-IF-PROFILE-EXISTS

User response:

No action is required.

CKZ8B407I **EXPORT command PROFILE-CREATOR = *profile_creator*, PROFILE-NAME = *profile_name***

Explanation:

This message lists the profile that is currently being exported.

User response:

No action is required.

CKZ8B408I **The number of executed commands: *number_of_commands***

Explanation:

This message displays the number of commands that were run.

User response:

No action is required.

CKZ8B409I **Beginning of the {EXPORT command | IMPORT command} summary report | End of the {EXPORT command | IMPORT command} summary report**

Explanation:

This message precedes or follows the EXPORT or IMPORT command summary.

User response:

No action is required.

CKZ8B410I **{EXPORT command | IMPORT command} for *profile_type* cloning profiles completed successfully**

Explanation:

The command that is listed in the message successfully completed for the subsystem or table space cloning profiles.

User response:

No action is required.

CKZ8B411W **EXPORT command | IMPORT command for *profile_type* cloning profiles completed with *completion_type***

Explanation:

The command that is listed in the message completed with either warnings or errors for the type of profile that is listed in the message.

User response:

Check previous messages for more information.

CKZ8B412I **EXPORT command | IMPORT command for DB2 subsystems information completed successfully**

Explanation:

The command that is listed in the message successfully completed for the DB2 subsystem information.

User response:

No action is required.

CKZ8B413W **EXPORT command | IMPORT command for DB2 subsystems information completed with *completion_type***

Explanation:

The command that is listed in the message completed with either warnings or errors for the DB2 subsystems.

User response:

Check previous messages for more information.

CKZ8B421E **The parameter *parameter_name* is already defined**

Explanation:

The parameter that is listed in the message was already defined.

User response:

Remove the extra parameter and resubmit the job.

CKZ8B422W **Unknown parameter '*unknown_parameter*'**

Explanation:

The parameter that is listed in the message is incorrect.

User response:

Correct or remove the parameter and resubmit the job.

CKZ8B423E **Invalid value was specified for PROFILE-TYPE**

Explanation:

An invalid value was specified for the PROFILE-TYPE parameter. Valid values are SS or TS.

User response:

Correct the parameter and resubmit the job.

CKZ8B424E **Not enough info to determine parameter files**

Explanation:

The parameter files must be specified using either DD names or data set and member names.

User response:

Specify either the DD names or the data set and member names for the CPARM and SPARM parameter files.

CKZ8B425E **Required parameter
parameter_name is not specified**

Explanation:

A required parameter was not specified.

User response:

Supply the missing parameter and resubmit the job.

CKZ8B426E **Mutually exclusive parameters
were specified for parameter file.
Either partitioned data sets or DD
names must be specified, but not
both**

Explanation:

Both partitioned data sets and DD names were specified in the job. Only one should be specified.

User response:

Remove either the partitioned data sets or the DD names and resubmit the job.

CKZ8B427E **A cloning parameter file (CPARM)
and a DB2 system parameter file
(SPARM) are identical**

Explanation:

Two different files must be specified for the CPARM and SPARM parameter files.

User response:

Specify different files for each and resubmit the job.

CKZ8B428E **The length of parameter value
parameter_name exceeds the
maximum length**

Explanation:

The parameter value of the parameter that is listed in the message is too long.

User response:

Correct the parameter and resubmit the job.

CKZ8B441E **Clone profiles are not defined for
export**

Explanation:

One or more profiles must be selected for export.

User response:

Enter the PROFILES, PROFILE-LIKE, or CREATOR-LIKE keywords to specify the profiles that you want to export.

CKZ8B442E **Profile name required for export**

Explanation:

A syntax error was found in the PROFILES keyword. The profile name was not specified in the job.

User response:

Specify profiles using the format "*profile-creator*."*profile-name*" and resubmit the job.

CKZ8B443E **Profile creator required for export**

Explanation:

A syntax error was found in the PROFILES keyword. The profile creator was not specified in the job.

User response:

Specify profiles using the format "*profile-creator*."*profile-name*" and resubmit the job.

CKZ8B444E **Syntax error: expected '!' between
profile creator and profile name**

Explanation:

A syntax error was found in the PROFILES keyword. A period must separate the profile creator and profile name.

User response:

Edit the job and enter a period between the profile creator and profile name.

CKZ8B445E **Either the PROFILE-LIKE and
CREATOR-LIKE parameters or the
PROFILES parameter must be
specified**

Explanation:

One of the profile selection parameters must be specified.

User response:

Specify one of the profile parameters that is listed in the message.

CKZ8B446E **Mutually exclusive profile
selection parameters are
specified. Specify either the
PROFILE-LIKE and CREATOR-LIKE
parameters or the PROFILES
parameter**

Explanation:

Both types of the profile selection parameters were specified, but only one is permitted.

User response:

Specify only one type of the profile selection parameters that is listed in the message.

CKZ8B447E **Invalid value of PROFILES: the
length of *profile_name* |
profile_creator exceeds the
maximum length**

Explanation:

Incorrect profiles were specified in the PROFILES parameter.

Explanation:

Specify the correct profiles and resubmit the job.

CKZ8B448E **Invalid syntax of EXPORT
command**

Explanation:

An error occurred for the EXPORT command.

User response:

Check the documentation for the EXPORT command syntax and correct the syntax before resubmitting the job.

CKZ8B451E Invalid value was specified for IMPORT-ACTION-IF-SUBSYSTEM-EXISTS

Explanation

An invalid value was specified for the IMPORT-ACTION-IF-SUBSYSTEM-EXISTS keyword. Valid values are:

- SKIP: Skip the import of this DB2 subsystem data.
- REPLACE: Replace the existing DB2 subsystem data.

User response:

Correct the value and resubmit the job.

CKZ8B452E Invalid value was specified for IMPORT-ACTION-IF-PROFILE-EXISTS

Explanation

An invalid value was specified for the IMPORT-ACTION-IF-PROFILE-EXISTS keyword. Valid values are:

- SKIP: Skip the import of this profile.
- REPLACE: Replace the existing profile.
- NEW: Import the profile with a new generated name.

User response:

Correct the value and resubmit the job.

CKZ8B453E Invalid value was specified for USE-ORIGINAL-CREATORS

Explanation

An invalid value was specified for the USE-ORIGINAL-CREATORS keyword. Valid values are:

- Y: Use the original profile creator.
- N: Use the user ID that runs the IMPORT command as the profile creator.

User response:

Correct the value and resubmit the job.

CKZ8B461E Syntax error: either command or parameter expected

Explanation:

An error occurred due to incorrect syntax.

User response:

Review the documentation for command syntax. Correct the error and resubmit the job.

CKZ8B462E Syntax error: expected "" for parameter_name parameter

Explanation:

The value for the parameter that is listed in the message must be surrounded by double quotation marks.

User response:

Correct the parameter value and resubmit the job.

CKZ8B463E Syntax error: expected '(' for parameter_name parameter

Explanation:

An error occurred due to incorrect syntax.

User response:

Review the documentation for command syntax. Correct the error and resubmit the job.

CKZ8B464E Syntax error: either excess or missing "" for parameter_name parameter

Explanation:

An error occurred due to incorrect syntax.

User response:

Review the documentation for command syntax. Correct the error and resubmit the job.

CKZ8B471E CKZPRINT cannot be opened. system message

Explanation:

CKZPRINT is a required DD for message output.

User response:

Check the DD statement for CKZPRINT and ensure that the CKZPRINT DD can be opened. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ8B472E An error occurred while writing to CKZPRINT | CKZERROR system_message

Explanation:

An error occurred when writing to one of the DDs for messages output.

User response:

If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZ8B481E Unknown command 'unknown_command'

Explanation:

An error occurred due to incorrect syntax.

User response:

Review the documentation for command syntax.
Correct the error and resubmit the job.

CKZ8B491E Invalid syntax of MIGRATE command

Explanation:

An error occurred for the MIGRATE command.

User response:

Check the documentation for the MIGRATE command syntax and correct the syntax before resubmitting the job.

CKZ8B492E Invalid value was specified for SIMULATE

Explanation

An invalid value was specified for the SIMULATE keyword. Valid values are:

- Y: Simulate mode is enabled. No changes will be added into the profile repository.
- N: Simulate mode is disabled.

User response:

Correct the value and resubmit the job.

CKZ8B493E Invalid value was specified for PROFILE-TYPE

Explanation

An invalid value was specified for the PROFILE-TYPE keyword. Valid values are:

- TS: Table space clone profiles
- SS: Subsystem clone profiles

User response:

Correct the value and resubmit the job.

CKZ8B494E Invalid value was specified for MIGRATE-ACTION-IF-PROFILE-EXISTS

Explanation

An invalid value was specified for the MIGRATE-ACTION-IF-PROFILE-EXISTS keyword. Valid values are:

- SKIP: Skip the migration of this profile.
- REPLACE: Replace the existing profile.
- NEW: Migrate the profile with a new generated name.

User response:

Correct the value and resubmit the job.

CKZ90101E RENAME MASK IS INVALID: new name mask

Explanation:

New name mask is invalid.

User response:

Correct the error in the RENAME-MASKS keyword.

CKZ90102E NEW NAME GREATER THAN 44 BYTES: new datasetname

Explanation:

New name masks resulted in new data set name that was greater than 44 characters.

User response:

Correct the error in the RENAME-MASKS keyword.

CKZ90103E NEW NAME IS INVALID: new datasetname

Explanation:

New name masks resulted in new data set name that was invalid.

User response:

Correct the error in the RENAME-MASKS keyword.

CKZ90201W SMS CLASSES NOT RETURNED BY ACS ROUTINES: RC=yyyyyyyy REASON=zzzzzzzz DSN=datasetname

Explanation:

Attempt to derive the new SMS classes for a data set failed.

User response:

Check job log and CKZPRINT for any additional messages. If unable to determine the reason for failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ90207W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation:

An error occurred using CKZ01HEX to print a record. Processing continues.

User response:

Please report this message to IBM Software Support.

CKZ90410E UNEXPECTED RETURN CODE FROM RACROUTE

Explanation:

A RACROUTE macro call got an unexpected return code. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing containing this message.

CKZ96501E INIT call not done

Explanation:

Module CKZ00965 has been called in an invalid sequence. Processing terminates.

User response:

Verify that correct versions of Db2 Cloning Tool modules are being used. If unable to resolve the problem, contact IBM Software Support.

CKZ96502E LOAD failed for: *program_name*

Explanation:

The indicated program name was not found. Processing terminates.

User response:

Check that the load library allocations are correct. If unable to resolve the problem, contact IBM Software Support.

CKZ96503E DDNAME missing: *ddname*

Explanation:

A required DD with the DD name that is listed in the message is not currently allocated. Processing terminates.

User response:

Either correct the DD name that was specified, or add the appropriate DD name allocation.

CKZ96504E Fatal errors have occurred during //CKZINI processing

Explanation:

During program start, one or more problems occurred during decoding of the //CKZINI member. As the //CKZINI provides vital information for , the program cannot continue execution.

User response:

Contact IBM Software Support. Have available the listing that contains this message and the PARMLIB member that controls execution of Db2 Cloning Tool.

CKZ96507E Error calling CKZ01VV1 *internal_table_name* FUNCTION: function R15=*r15_contents* R0=*r0_contents*

Explanation:

A problem occurred using a dataspace. Processing terminates.

User response:

Contact IBM Software Support. Have available the CKZINI PARMLIB member that controls execution of Db2 Cloning Tool.

CKZ96508E Unknown INI token: *token* in section: *section*

Explanation:

An error occurred validating the CKZINI PARMLIB member options. Processing terminates.

User response:

Verify the correct CKZINI member in the Db2 Cloning Tool PARMLIB data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ96509E Invalid INI value for section: *section* token: *token* value: *value*

Explanation:

An error occurred validating the CKZINI PARMLIB member options. Processing terminates.

User response:

Correct the CKZINI member in the Db2 Cloning Tool PARMLIB data set. If unable to resolve the problem, contact IBM Software Support.

CKZ96590E Unsupported function: *function*

Explanation:

Module CKZ00965 has been called to perform an unknown function. Processing terminates.

User response:

Verify the correct version of module CKZ00965 is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ98001E Subtask *subtask_number*, RS=*reason_code*, invalid MIB

Explanation:

This is an internal error. The input to a routine is invalid.

User response:

Contact IBM Software Support.

CKZ98002E Subtask *subtask_number*, RS=*reason_code*, CEEPIPI LOAD failed

Explanation:

The LE load module CEEPIPI must be available to load when DATA-MASKING(Y) is specified in the COPY command.

User response:

Make the LE library available at run time and resubmit the job.

CKZ98003E Subtask *subtask_number*, RC=*return_code*, RS=*reason_code*, TOKEN=*token_value_in_use*, LERC=*le_return_code*, LERS=*le_reason_code*, LEFDBK=*le_feedback_code*, LE interface error, call type=*call_type*

Explanation:

This is an error return from the LE interface. *call_type* can be one of the following: INIT, STARTSEQ, CALL990, ENDSEQ, ENDL.

User response:

Contact IBM Software Support.

CKZ98004E Subtask *subtask_number*,
RC=*return_code*, RS=*reason_code*,
TOKEN=*token_value_in_use*, error
return from the data masking
processor

Explanation:

This is an error returned from the data masking processor. Data masking is terminated.

User response:

Contact IBM Software Support.

CKZ98101E Subtask *subtask_number*,
RS=*reason_code*, invalid MIB

Explanation:

This is an internal error. The input to a routine is invalid.

User response:

Contact IBM Software Support.

CKZ98102E Subtask *subtask_number*,
RS=*reason_code*, *module_name*
LOAD failed

Explanation:

This is an internal error. The LOAD macro failed to load the indicated module. The *reason_code* shows the reason code from the LOAD macro invocation.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ98103E Subtask *subtask_number*,
RC=*return_code*, RS=*reason_code*,
TOKEN=*LE_token*,
LERC=*LE_return_code*, LERS=
LE_reason_code,
LEFDBK=*LE_feedback_code*,
CallType=*LE_call_type*

Explanation

An error occurred during the call of the Language Environment(LE) function. The *LE_call_type* is one of the following:

- INIT is the call of *init_sub_dp*.
- STARTSEQ is the call of *start_seq*.
- ENDSEQ is the call of *end_seq*.
- ENDLE is the call of *term*.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ98104E Subtask *subtask_number*,
RC=*return_code*, RS=*reason_code*,
TOKEN=*token_value*, error return
from the rule processor

Explanation:

An error was returned from the data masking processor. Data masking is terminated.

User response:

Contact IBM Software Support.

CKZ98401E Subtask *subtask_number*,
RS=*reason_code*, Invalid VIB

Explanation:

This is an internal error. The VIB is incorrect.

User response:

Contact IBM Software Support.

CKZ98402E Subtask *subtask_number*,
RS=*reason_code*, *module_name*
LOAD failed

Explanation:

This is an internal error. The LOAD macro failed to load the indicated module. The *reason_code* shows the reason code from the LOAD macro invocation.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ98403E Subtask *subtask_number*,
RC=*return_code*, RS=*reason_code*,
TOKEN=*LE_token*, Error return
from the CKZ9DB2D module

Explanation:

An error occurred during the load of the CKZ9DB2D module.

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ98404E Subtask *subtask_number*,
RC=*return_code*, RS=*reason_code*,
TOKEN=*LE_token*,
LERC=*LE_return_code*, LERS=
LE_reason_code, LEFDBK=
LE_feedback_code,
CallType=*LE_call_type*

Explanation

An error occurred during the call of the Language Environment(LE) function. The *LE_call_type* is one of the following:

- INIT is the call of *init_sub_dp*
- STARTSEQ is the call of *start_seq*
- ENDSEQ is the call of *end_seq*
- ENDLE is the call of *term*

User response:

If unable to resolve this error, contact IBM Software Support.

CKZ98421E Subtask *subtask_number*, Wrong CKZ9DB2D module version. Expected version *module_version* but found version *module_version*

Explanation:

This is an internal error. The version of the CKZ9DB2D module does not match the expected version.

User response:

Contact IBM Software Support.

CKZ99000I SUBTASK *subtask_number*, DATA MASKING ENGINE HAS INITIALIZED SUCCESSFULLY

Explanation:

This message is informational.

User response:

No action is required.

CKZ99001E SUBTASK *subtask_number*, UNABLE TO INITIALIZE DATA MASKING ENGINE

Explanation:

An error occurred during initialization of data masking engine.

User response:

See previous error messages for more information.

CKZ99002E SUBTASK *subtask_name*, UNABLE TO FIND FIELD *field_name* IN THE TABLE *creator_name.table_name*

Explanation:

The field doesn't exist in the table. This error happens during RI or masks initialization.

User response:

Contact IBM Software Support if this error happens during RI initialization. Change field name in the mask rule definition if this error happens during masks initialization.

CKZ99003E SUBTASK *subtask_number*, UNABLE TO ADD LINK BETWEEN *parent_creator_name.parent_table_name* (*parent_field_name*) AND *child_creator_name.child_table_name* (*child_field_name*)

Explanation:

An error occurred during internal adding of link between tables.

User response:

See previous error messages for more information.

CKZ99004E SUBTASK *subtask_number*, MASK RULE *mask_rule-name* IS NOT

ALLOWED FOR FIELD *creator_name.table_name(field_name)* WHICH HAS TYPE *field_type*

Explanation:

A mask rule with designated name is not applicable for the type of designated field.

User response:

Change mask rule or field name in the mask rule definition.

CKZ99005E SUBTASK *subtask_number*, AN ERROR OCCURRED WHILE ADDING *mask_rule_name* FOR *creator_name.table_name(field_name)*

Explanation:

An error occurred during internal adding of mask rule.

User response:

See previous error message for more information.

CKZ99006E SUBTASK *subtask_number*, *parameter_name* VALUE IS OUT OF RANGE [*minimum_value*, *maximum_value*]; RULE='*mask_rule_name*', FIELD='*creator_name.table_name(field_name)*'

Explanation:

Parameter value is out of range.

User response:

Change parameter value in the mask rule definition.

CKZ99007E SUBTASK *subtask_number*, MINIMAL (*minimum_parameter*) VALUE EXCEEDED OR EQUALED TO MAXIMAL (*maximum_parameter*) VALUE

Explanation:

A MINIMAL value exceeded or equaled the MAXIMAL value.

User response:

Change MINIMAL or MAXIMAL value of range in the mask rule definition.

CKZ99008E SUBTASK *subtask_number*, DESIGNATED RANGE [*minimum_parameter*, *maximum_parameter*] IN MASK RULE EXCEEDED FIELD LENGTH (*field_length*)

Explanation:

The designated range in the MASK RULE exceeded field length.

User response:

Change range in the mask rule definition.

CKZ99009E SUBTASK *subtask_number*,
UNABLE TO LOAD USER EXIT
user_exit_name

Explanation:

Unable to load user exit.

User response:

Change the user exit name in the mask rule definition.

CKZ99010E SUBTASK *subtask_number*,
UNABLE TO PARSE SUPPLIED
PATTERN '*pattern*'

Explanation:

Unable to parse supplied mask pattern.

User response:

See previous error messages for more information.

CKZ99011I SUBTASK *subtask_number*,
ADDING LINK BETWEEN
parent_creator_name.parent_table
_name
(*parent_field_name*) AND
child_creator_name.child_table_na
me
(*child_field_name* COMPLETED
SUCCESSFULLY

Explanation:

This message is informational.

User response:

No action is required.

CKZ99012I SUBTASK *subtask_number*,
ADDING *mask_rule* FOR
creator_name.table_name(field_na
me) COMPLETED SUCCESSFULLY

Explanation:

This message is informational.

User response:

No action is required.

CKZ99013E SUBTASK *subtask_number*,
UNABLE TO FIND TABLE
creator_name.table_name

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99014E SUBTASK *subtask_number*,
UNABLE TO ADD CREATOR
creator_name

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99015E SUBTASK *subtask_number*, FIELD
field_name ALREADY EXISTS IN
creator_name.table_name

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99016E SUBTASK *subtask_number*, TABLE
creator_name.table_name
ALREADY EXISTS

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99017E SUBTASK *subtask_number*,
EXPECTED SYMBOL *symbol* NOT
RECEIVED

Explanation:

The expected symbol was not received in a mask pattern.

User response:

See error messages following this error for more information.

CKZ99018E SUBTASK *subtask_number*,
INVALID MASK PATTERN
'*mask_pattern*'

Explanation:

An invalid mask pattern was encountered.

User response:

See error messages following this error for more information.

CKZ99019E SUBTASK *subtask_number*,
UNEXPECTED STRING '*string*' HAS
BEEN RECEIVED

Explanation:

An unexpected string was encountered in a mask pattern.

User response:

See error messages following this error for more information.

CKZ99020E SUBTASK *subtask_number*, THE
SPECIFIED STRING '*error_string*'
DOES NOT CORRESPOND TO
mask_rule_name PATTERN

Explanation:

The specified string not corresponded to a mask pattern.

User response:

Correct the string in the mask rule definition.

CKZ99021E SUBTASK *subtask_number*,
UNABLE TO FIND CREATOR
creator_name

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99022E SUBTASK *subtask_number*,
INVALID MASK RULE NAME:
'*mask_rule_name*'

Explanation:

An invalid mask rule name was entered.

User response:

Correct the mask rule name in the mask rule definition.

CKZ99023E SUBTASK *subtask_number*,
UNABLE TO GET CURRENT
SYSTEM DATE

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99024E SUBTASK *subtask_number*,
UNABLE TO GET CURRENT
SYSTEM TIME

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99025E SUBTASK *subtask_number*,
UNABLE TO GET CURRENT
SYSTEM TIMESTAMP

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99026E SUBTASK *subtask_number*,
DESIGNATED KEY
parent_creator_name.parent_table_name
(*parent_field_name*) <=>
child_creator_name.child_table_name
(*child_field_name*) ALREADY
EXISTS

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99027E SUBTASK *subtask_number*,
UNABLE TO ADD RULE '*mask_rule*'
BECAUSE FIELD
creator_name.table_name(field_name)
INVOLVED IN RELATION

Explanation:

Unable to add the rule because the field listed is involved in relation.

User response:

Delete mask rule definition or change the field name in the mask rule definition.

CKZ99028E SUBTASK *subtask_number*,
UNABLE TO GET CURRENT
SYSTEM USER

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99029E SUBTASK *subtask_number*,
UNABLE TO ADD TABLE
creator_name.table_name

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99030E SUBTASK *subtask_number*,
UNABLE TO ADD FIELD
creator_name.table_name(field_name)

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99031E SUBTASK *subtask_number*,
INVALID FIELD TYPE: '*field_type*'

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99032E SUBTASK *subtask_number*,
INVALID MIE_RECfmt =
value_of_MIE_RECfmt. 9 IS
EXPECTED (__RFPAGE -
PROCESS PAGE)

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99033E SUBTASK *subtask_number*,
POINTER TO THE MIE STRUCT IS
NULL

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99034E SUBTASK *subtask_number*,
POINTER TO THE
DATA_PAGE_HEADER STRUCT IS
NULL

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99035E SUBTASK *subtask_number*,
CURRENT PAGE SIZE IS ZERO

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99036E SUBTASK *subtask_number*,
PDSN_FORMAT IS NOT DEFINED

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99037E SUBTASK *subtask_number*,
INVALID PDSN_FORMAT =
'*pdsn_format_symbol*'. ' ' OR 'R' IS
EXPECTED

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99038W SUBTASK *subtask_number*,
PGFLAGS =
hex_value_of_PGFLAGS. CURRENT
DATA PAGE WAS NOT PROCESSED

Explanation:

The current data page was not processed because of a problem with PGFLAGS. Therefore, not all data was processed in the data base. PGFLAGS is internal hex value.

User response:

If this error reoccurs and you would like this data page to be processed, contact IBM Software Support.

CKZ99039W SUBTASK *subtask_number*,
BPAGELASTBYTEFLAG =
hex_value_of_bPageLastByteFlag;
BTPAGELASTBYTE =
hex_value_of_btPageLastByte .
CURRENT DATA PAGE WAS NOT
PROCESSED

Explanation:

The current data page was not processed due to problem with the pairing of bPageLastByteFlag and btPageLastByte. It must be: 1) bPageLastByteFlag = zero, btPageLastByte = 0xC5; or 2) bPageLastByteFlag = not zero, btPageLastByte = 0xD5. Other combinations are invalid. If that is the case, then not all data was processed in the data base.

User response:

If masking only a portion of the data is unacceptable, contact IBM Software Support.

CKZ99040W SUBTASK *subtask_number*, Page
record was not processed: offset =
hex_value_of_offset, PGSFLAGS =
hex_value_of_PGSFLAGS
[*flags_descr*]

Explanation:

The specified page record does not meet data masking restrictions. The mask could not be applied to the page.

User response:

If unable to resolve this warning, contact IBM Software Support. Have available the listing that contains this message.

CKZ99041E SUBTASK *subtask_number*, TABLE
NOT FOUND BY OBID = *obid*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99042I SUBTASK *subtask_number*,
CURRENT "PAGE" CALL
PROCESSED SUCCESSFULLY

Explanation:

This message is informational.

User response:

No action is required.

CKZ99043W SUBTASK *subtask_number*,
CURRENT "PAGE" CALL
COMPLETED WITH WARNINGS

Explanation:

Warnings have been issued during processing of "PAGE" call.

User response:

See previous warning messages.

CKZ99044E **SUBTASK *subtask_number*,
UNABLE TO PROCESS THE
CURRENT "PAGE" CALL DUE TO
ERRORS**

Explanation:

An error occurred during processing of "PAGE" call.

User response:

See previous error messages.

CKZ99045I **SUBTASK *subtask_number*,
CURRENT "SDSN" CALL
PROCESSED SUCCESSFULLY**

Explanation:

This message is informational.

User response:

No action is required.

CKZ99046W **SUBTASK *subtask_number*,
CURRENT "SDSN" CALL
COMPLETED WITH WARNINGS**

Explanation:

Warnings have been issued during processing of "SDSN" call.

User response:

See previous warning messages.

CKZ99047E **SUBTASK *subtask_number*,
UNABLE TO PROCESS THE
CURRENT "SDSN" CALL DUE TO
ERRORS**

Explanation:

An error occurred during processing of "SDSN" call.

User response:

See previous error messages.

CKZ99048E **SUBTASK *subtask_number*,
INVALID MIE_RECFMT =
value_of_MIE_RECFMT. 8 IS
EXPECTED (__RFPDSN - PROCESS
DATA SET)**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99049I **SUBTASK *subtask_number*,
CURRENT DATA SET IS
*data_set_name***

Explanation:

This message is informational.

User response:

No action is required.

CKZ99050W **SUBTASK *subtask_number*,
INITIALIZATION COMPLETED
WITH WARNINGS**

Explanation:

Warnings occurred during initialization of data masking engine.

User response:

See previous warning messages.

CKZ99051E **SUBTASK *subtask_number*,
FTR_STATIC() FAULT**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99052E **SUBTASK *subtask_number*,
FTR_CURRENT_USER() FAULT**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99053E **SUBTASK *subtask_number*,
FTR_USER_EXIT() FAULT**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99054E **SUBTASK *subtask_number*,
FTR_SCRAMBLE() FAULT**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99055E **SUBTASK *subtask_number*,
MASK() FAULT**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99056E **SUBTASK *subtask_number*,
PATTERN() FAULT**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99057E **SUBTASK *subtask_number*,
INVALID MIE_COL_NULLS =**

**'value_of_MIE_COL_NULLS'. IT
MUST BE A 'Y' OR 'N'**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99058E SUBTASK *subtask_number*,
INVALID MASK TYPE
'*mask_rule_type*' FOR FIELD
*creator_name.table_name(field_na
me)*

Explanation:

Mask with designated type is not applicable for the designated field.

User response:

Correct the mask rule or field name in the mask rule definition.

CKZ99059E SUBTASK *subtask_number*,
INVALID MASK PATTERN
'*mask_pattern_string*' FOR FIELD
*creator_name.table_name(field_na
me)*

Explanation:

Designated mask pattern is not applicable for designated field.

User response:

Correct the mask rule or field name in the mask rule definition.

CKZ99060E SUBTASK *subtask_number*, RI
INITIALIZATION ERROR HAS
OCCURRED

Explanation:

Initialization has completed with RI initialization error.

User response:

See previous error message for more information.

CKZ99061E SUBTASK *subtask_name*, INVALID
MIE_RECfmt =
value_of_MIE_RECfmt. 3 IS
EXPECTED (__RFIDSN - INITIAL
DATA SET)

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99062E SUBTASK *subtask_name*, INVALID
MIE_RECfmt =
value_of_MIE_RECfmt. 4 IS
EXPECTED (__RFTS - TABLE
SPACE).

**RFIDSN.MIE_IDSN_DSNAME =
"data_set_name"**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99063E SUBTASK *subtask_name*, INVALID
MIE_RECfmt =
value_of_MIE_RECfmt. 5 IS
EXPECTED (__RFTB - TABLE
RECORD).
RFIDSN.MIE_IDSN_DSNAME =
"data_set_name"

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99064W SUBTASK *subtask_name*, LENGTH
OF VALUE OF PATTERN '*pattern*' IS
GREATER THAN LENGTH OF
COLUMN
*creator_name.table_name(field_na
me)*

Explanation:

The value of pattern mask will be truncated in target column.

User response:

Change the mask rule or field name in the mask rule definition. You may leave the mask rule untouched but note that its value will be truncated.

CKZ99065E SUBTASK *subtask_name*, INVALID
MIE_RECfmt =
value_of_MIE_RECfmt. 7 IS
EXPECTED (__RFRI -
REFERENTIAL INTEGRITY).
RFIDSN.MIE_IDSN_DSNAME =
"data_set_name" TABLE NAME =
"creator_name.table_name"

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99066E SUBTASK *subtask_number*,
PARENT TABLE
"*parent_creator_name.parent_tabl
e_name*" INVOLVED IN RELATION
IS NOT FOUND
RFIDSN.MIE_IDSN_DSNAME =
"data_set_name" RI NAME =
"*relation_name*" CHILD TABLE
NAME =

"*child_creator_name.child_table_name*"

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99067E SUBTASK *subtask_name*, CHILD TABLE HAS RI FOR PARENT TABLE BUT PARENT TABLE HAS NOT RI FOR CHILD TABLE CHILD TABLE NAME = "*child_creator_name.child_table_name*" PARENT TABLE NAME = "*parent_creator_name.parent_table_name*" RI NAME = "*relation_name*" RFIDSN.MIE_IDSN_DSNAME = "*data_set_name*"

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99068E SUBTASK *subtask_name*, RFKEY IS NOT FOUND IN PARENT RFRI CHILD TABLE NAME = "*child_creator_name.child_table_name*" PARENT TABLE NAME = "*parent_creator_name.parent_table_name*" RI NAME = "*relation_name*" RFIDSN.MIE_IDSN_DSNAME = "*data_set_name*"

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99069E SUBTASK *subtask_number*, RFKEY IS NOT FOUND IN CHILD RFRI CHILD TABLE NAME = "*child_creator_name.child_table_name*" PARENT TABLE NAME = "*parent_creator_name.parent_table_name*" RI NAME = "*relation_name*" RFIDSN.MIE_IDSN_DSNAME = "*data_set_name*"

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99070I SUBTASK *subtask_number*, CURRENT SUBTASK HAS BEEN INITIALIZED

Explanation:

This message is informational.

User response:

No action is required.

CKZ99071I SUBTASK *subtask_number*, CURRENT SUBTASK HAS BEEN TERMINATED

Explanation:

This message is informational.

User response:

No action is required.

CKZ99072I SUBTASK *subtask_number*, DATA MASKING ENGINE HAS BEEN TERMINATED SUCCESSFULLY

Explanation:

This message is informational.

User response:

No action is required.

CKZ99073W SUBTASK *subtask_number*, TERMINATION OF DATA MASKING ENGINE HAS COMPLETED WITH WARNINGS

Explanation:

This message says that there were warnings during termination of data masking engine.

User response:

See previous warning messages.

CKZ99074E SUBTASK *subtask_number*, UNABLE TO TERMINATE THE DATA MASKING ENGINE

Explanation:

An error occurred during termination of data masking engine.

User response:

See previous error messages for more information.

CKZ99075E SUBTASK *subtask_number*, FATAL ERROR - SEMGET() FAILED

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99076I SUBTASK *subtask_number*, PRIVATE RANDOM GENERATOR SEMAPHORE HAS BEEN CREATED SUCCESSFULLY

Explanation:

This message is informational.

User response:

No action is required.

CKZ99077E **SUBTASK *subtask_number*, FATAL ERROR - CANNOT INITIALIZE THE RANDOM GENERATOR SEMAPHORE**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99078W **SUBTASK *subtask_number*, CANNOT REMOVE THE RANDOM GENERATOR SEMAPHORE**

Explanation:

This is an internal warning. It does not affect the data producing by data masking.

User response:

No action is required.

CKZ99079I **SUBTASK *subtask_number*, RANDOM GENERATOR SEMAPHORE HAS BEEN REMOVED SUCCESSFULLY**

Explanation:

This message is informational.

User response:

No action is required.

CKZ99080E **SUBTASK *subtask_number*, FATAL ERROR - LOCKING OF RANDOM GENERATOR SEMAPHORE HAS FAILED**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99081E **SUBTASK *subtask_number*, FATAL ERROR - UNLOCKING OF RANDOM GENERATOR SEMAPHORE HAS FAILED**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ99082E **Subtask *subtask_number*, Invalid range of *parameter_name*: [*minimum_value*, *maximum_value*]**

Explanation:

Invalid range of parameter in mask rule, i.e. minimum value exceeded the maximum value.

User response:

See the next error messages to determine the incorrect mask rule and correct the range.

CKZ99083E **Subtask *subtask_number*, (MIN_TIMESTAMPZ + MIN_TIMEZONE) is out of range**

Explanation:

Calculated minimum bound of TIMESTAMP WITH TIME ZONE is out of allowed range for this type of column in the RANDOM mask rule.

User response:

Change MIN_TIMESTAMPZ and/or MIN_TIMEZONE parameters in the RANDOM mask rule, shown in the error messages following this error.

CKZ99084E **Subtask *subtask_number*, (MAX_TIMESTAMPZ + MAX_TIMEZONE) is out of range**

Explanation:

Calculated maximum bound of TIMESTAMP WITH TIME ZONE is out of allowed range for this type of column in the RANDOM mask rule.

User response:

Change MAX_TIMESTAMPZ and/or MAX_TIMEZONE parameters in the RANDOM mask rule shown in error messages following this error.

CKZ99085E **SUBTASK *subtask_number*, Unable to get current system time zone**

Explanation:

This is an internal error.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ99086E **SUBTASK *subtask_number*, Data masking engine does not support current release of Db2: *Db2_release_number***

Explanation:

The data masking engine does not support the current release of Db2.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ99087E **Subtask *subtask_number*, iconv_open() error: Reason: *reason_code*; From CCSID: *from_CCSID*; To CCSID: *to_CCSID***

Explanation

String column data conversion error was issued. C function iconv_open() has failed. reason_code may be:

- EINVAL: the conversion between encoded character sets specified is not supported
- ECUNNOENV: a CUN_RS_NO_UNI_ENV error was issued by Unicode Conversion Services
- ECUNNOCONV: a CUN_RS_NO_CONVERSION error was issued by Unicode Conversion Services
- ECUNNOTALIGNED : a CUN_RS_TABLE_NOT_ALIGNED error was issued by Unicode Conversion Services
- ECUNERR: function iconv_open() encountered an unexpected error while using Unicode Conversion Services

User response:

Contact IBM Software Support.

CKZ99088E Subtask *subtask_name*, iconv() error: Reason: *reason_code*; From CCSID: *from_CCSID*; To CCSID: *to_CCSID*

Explanation

String column data conversion error was issued. C function iconv() has failed. *reason_code* may be:

- EBADF: conversion descriptor is not valid
- ECUNNOENV: a CUN_RS_NO_UNI_ENV error was issued by Unicode Conversion Services
- ECUNNOCONV: a CUN_RS_NO_CONVERSION error was issued by Unicode Conversion Services
- ECUNNOTALIGNED: a CUN_RS_TABLE_NOT_ALIGNED error was issued by Unicode Conversion Services
- ECUNERR: function iconv() encountered an unexpected error while using Unicode Conversion Services
- EILSEQ: input conversion stopped due to an input byte that does not belong to the input codeset
- EINVAL: input conversion stopped due to an incomplete character or shift sequence at the end of the input buffer
- E2BIG: input conversion stopped due to lack of space in the output buffer

User response:

Contact IBM Software Support.

CKZ99089E Subtask *subtask_number*, Unable to add rule '*mask_rule*' because CCSID is not defined for column *creator_name.table_name(column_name)*. Reason: *reason_code*

Explanation

CCSID is not defined for string column because it was created prior to Version 8. The data masking module attempted to retrieve the CCSID from SYSIBM.SYSTABLESPACE but encountered an error.

reason_code may be:

- 1: string column contains binary data and does not have valid CCSID
- 2: subtype of string column is not valid
- 3: database was created in a DB2 release prior to Version 5

User response

If *reason_code* = 1 then

1. Remove mask rule from list of mask rules
2. Change the mask rule to another column, or to the USEREXIT mask rule.

If *reason_code* = 2 or 3, contact IBM Software Support.

CKZ99090E Subtask *subtask_number*, Unable to add rule '*mask_rule*' because string column *creator_name.table_name(column_name)* contains binary data

Explanation:

Unable to add defined mask rule to string column which contains binary data.

User response:

Remove mask rule from list of mask rules, or change the mask rule to another column, or to the USEREXIT mask rule.

CKZ99091W Subtask *subtask_number*, Decimal floating-point value *decimal_floating_point_value_from_mask_rule* is out of range in *mask_rule_name* rule for column *creator_name.table_name(column_name)*. It is converted to *special_value*

Explanation

The defined decimal floating-point value is out of range. The value was converted to a *special_value*, which can be one of the following:

- +INFINITY
- -INFINITY
- +ZERO (+0E0)
- -ZERO (-0E0)

User response:

To eliminate this message, change the decimal floating-point value in the mask rule so it is in the range of the current decimal floating-point format.

CKZ99092I **Subtask** *subtask_number*,
load_module_name *build_date*
build_time *VERS=version* *fff*

Explanation:

This message is informational and provides the module level. The *fff* fields is currently unused.

User response:

No action is required.

CKZ99093W **SUBTASK** *subtask_number*, Unable
to interpret XML data: Page
(PGNUM):
hex_value_of_page_number, Table
record offset:
hex_value_of_record_offset, XML
data column offset:
hex_value_of_xml_data_column_off
set, Node kind offset:
offs_of_curr_node_kind_value

Explanation:

The data masking engine was unable to interpret XML data.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ99094W **SUBTASK** *subtask_number*, Unable
to translate source string ID to
target string ID: Page (PGNUM):
hex_value_of_page_number, String
ID offset:
hex_value_of_string_id_offset,
String ID value:
hex_value_of_string_id_value

Explanation:

The data masking engine was unable to translate an XML string ID.

User response:

See previous error messages for more information. If unable to resolve this problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ99095E **SUBTASK** *subtask_number*, String
ID translating routine has failed:
MIB_EXIT_RS:
hex_value_of_mib_exit_rs, Page
(PGNUM):
hex_value_of_page_number, String
ID offset:
hex_value_of_string_id_offset,
String ID value:
hex_value_of_string_id_value

Explanation:

This is an internal error.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ99096W **SUBTASK** *subtask_number*, XML
record header interpreted
incorrectly: Page (PGNUM):
hex_value_of_page_number, Table
record offset:
hex_value_of_record_offset, XML
data column offset:
hex_value_of_xml_data_column_off
set, Node kind:
hex_value_of_node_kind, Node
kind offset:
hex_value_of_node_kind_offset,
Interpreted offset:
hex_value_of_interpreted_offset

Explanation:

The data masking engine was unable to interpret the XML record header.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ99097W **SUBTASK** *subtask_number*,
Unsupported combination of two
tables with identical OBID *obid*
was identified. 1st table:
table_name1, **2nd table:**
table_name2

Explanation:

The data masking engine was called for two tables with the same table OBIDs that are not involved in a clone relationship. Table *table_name2* will not be processed by data masking.

User response:

See previous error messages for more information. If unable to resolve this problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ99098W **SUBTASK** *subtask_number*, **Two**
tables with identical OBID *obid*
and TYPE *type_value* **were**
identified. 1st table: *table_name1*,
2nd table: *table_name2*

Explanation:

The data masking engine was called for two tables with the same table OBIDs and the same type. Table *table_name2* will not be processed by data masking.

User response:

See previous error messages for more information. If unable to resolve this problem, contact IBM Software

Support. Have available the listing that contains this message.

CKZ99099E **SUBTASK *subtask_number*,
Incorrect current data set name:
*data_set_name***

Explanation:

This is an internal error.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZ990100W **SUBTASK *subtask_number*, Table
with OBID *obid* which belongs to
Base & Clone tables was identified
| Identified table: *it_value*, Base
table: *bt_value*, Clone table:
*ct_value***

Explanation:

The data masking engine was called for a table with the same OBIDs that were already specified for a base/clone table pair.

User response:

See previous error messages for more information. If unable to resolve this problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ990101W **Subtask *subtask_number*, Unable
to add *mask_rule* for
creator.table(column) column
because it is part of the hash key**

Explanation:

The column that was listed in the message cannot be masked because it is defined as a part of the hash key.

User response:

Exclude this column from the masking and resubmit the job.

CKZ990102W **SUBTASK *subtask_number*, Unable
to add *mask_value* for
*creator_name.table_name(field_na
me)* column because it is related to
a hash key by RI**

Explanation:

Data masking cannot be applied to the specified column because it is a part of hash key.

User response:

Delete the mask rule definition or change the field name in the mask rule definition.

CKZ99107E **MINIMAL (*minimum_parameter*)
VALUE EXCEEDED OR EQUALED TO
MAXIMAL (*maximum_parameter*)
VALUE**

Explanation:

MINIMAL value exceeded or equaled to MAXIMAL value.

User response:

Change MINIMAL or MAXIMAL value of range in the mask rule definition.

CKZ99109E **UNABLE TO LOAD USER EXIT
*user_exit_name***

Explanation:

Unable to load user exit.

User response:

Change user exit name in the mask rule definition.

CKZ99110E **UNABLE TO PARSE SUPPLIED
PATTERN '*pattern*'**

Explanation:

Unable to parse supplied mask pattern.

User response:

See previous error messages for more information.

CKZ99117E **EXPECTED SYMBOL '*symbol*' NOT
RECEIVED**

Explanation:

Expected symbol not received in mask pattern. sss = symbol

User response:

See error messages following this error for more information.

CKZ99118E **INVALID MASK PATTERN
'*mask_pattern*'**

Explanation:

Invalid mask pattern.

User response:

See error messages following this error for more information.

CKZ99119E **UNEXPECTED STRING '*string*' HAS
BEEN RECEIVED**

Explanation:

An unexpected string has been received in mask pattern.

User response:

See error messages following this error for more information.

CKZ99120E **THE SPECIFIED STRING
'*error_string*' DOES NOT
CORRESPOND TO
mask_rule_name PATTERN**

Explanation:

The specified string does not correspond to mask pattern.

User response:

Correct the string in the mask rule definition.

CKZ99122E **INVALID MASK RULE NAME:**
'mask_rule_name'

Explanation:

Invalid mask rule name.

User response:

Correct the mask rule name in the mask rule definition.

CKZ99182E **Invalid range of parameter_name:**
[minimum_value, maximum_value]

Explanation:

Invalid range of parameter in mask rule; i.e. minimum value exceeded maximum value.

User response:

See the next error messages to determine the incorrect mask rule and correct the range.

CKZ99183E **(MIN_TIMESTAMPZ +**
MIN_TIMEZONE) is out of range

Explanation:

Calculated minimum bound of TIMESTAMP WITH TIME ZONE is out of the allowed range for this type of column in the RANDOM mask rule.

User response:

Change MIN_TIMESTAMPZ and/or MIN_TIMEZONE parameters in the RANDOM mask rule shown in error messages following this error.

CKZ99184E **(MAX_TIMESTAMPZ +**
MAX_TIMEZONE) is out of range

Explanation:

Calculated maximum bound of TIMESTAMP WITH TIME ZONE is out of allowed range for this type of column in the RANDOM mask rule.

User response:

Change MAX_TIMESTAMPZ and/or MAX_TIMEZONE parameters in the RANDOM mask rule shown in error messages following this error.

CKZ99206I **Unable to open DDNAME = ddname**
for debug output.

Explanation:

The DD name listed in the message could not be opened for debug output. Check accompanying z/OS messages for additional information.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing containing these messages.

CKZ99224E **Unable to add unique key**
unique_key(column_name).

Explanation:

Insufficient memory was available to allocate the unique key listed in the message. Check accompanying z/OS messages for additional information.

User response:

If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing containing these messages.

CKZ99225E **Unique key**
unique_key(column_name already
exists).

Explanation:

The name provided for the unique key already exists.

User response:

Change the unique key name and regenerate the DDL.

CKZ99246I **module_name build_date**
build_time VERS=version
REV=revision N/A build_date.

Explanation:

This message is used for troubleshooting purposes and provides information about the program and module.

User response:

No action is required.

CKZ9DB2D00E **Subtask subtask_number, Zero**
page size was detected in input
parameters

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D01E **Subtask subtask_number, Cannot**
open data set: data_set_name.

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D02E **Subtask subtask_number, Cannot**
close data set: data_set_name.

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ9DB2D03E Subtask *subtask_number*, Zero CI size was determined. Data set: *data_set_name*.

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D04E Subtask *subtask_number*, Page size is not multiple to CI size. Data set: *data_set_name*, Page size: *page_size*, CI size: *control_interval_size*

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D05E Subtask *subtask_number*, Zero CI count was determined. Data set: *data_set_name*

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D06E Subtask *subtask_number*, CI count is not multiple to number of CIs per page. Data set: *data_set_name*, CI count: *control_intervals_per_data_set*, Number of CIs per page: *control_intervals_per_page*

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D07E Subtask *subtask_number*, Cannot read page. Data set: *data_set_name*, RBA: *RBA_hex_value*, Page size: *page_size*

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D08E Subtask *subtask_number*, Internal error: For nonpartitioned page set the data set number is out of range. Data set number: *data_set_number*, Data set name [template]: *data_set_name_or_template*, Page number: (*page_number* | not calculated), Pages per data set: (*pages_per_data_set* | not calculated)

Explanation:

This is an internal error.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D09E Subtask *subtask_number*, Incorrect PGFLAGS in page header. Data set: *data_set_name*, Page number: *page_number*, Page size: *page_size*, PGFLAGS: *PGFLAGS_hex_value*.

Explanation:

An incorrect page header PGFLAGS was encountered.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D10E Subtask *subtask_number*, Page was saved incorrectly. Data set: *data_set_name*, Page number: *page_number*, Page size: *page_size*, PGCOMB last byte bit: *PGCOMB_last_byte_bit_hex_value*, Page last byte: *page_last_byte_hex_value*.

Explanation:

An incorrectly saved page was encountered.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D11E Subtask *subtask_number*, Incorrect header page PGFLAGS. Data set: *data_set_name*, PGFLAGS: *PGFLAGS_hex_value*.

Explanation:

A header page with incorrect PGFLAGS was encountered.

User response:

Use other messages from the cloning job output and refer to IBM documentation to attempt to resolve the problem, and submit the cloning job again. If unable to resolve the problem, contact IBM Software Support.

CKZ9DB2D12E Subtask *subtask_number*, Cannot allocate DD. Data set: *data_set_name*, *dynalloc()* error code: *dynalloc_c_function_error_code*, *dynalloc()* info code: *dynalloc_c_function_info_code*.

Explanation:

The DD for the listed data set cannot be allocated.

User response:

Find and eliminate the reason of the DD allocation problem by using *dynalloc_c_function_error_code* and *dynalloc_c_function_info_code*, then resubmit the cloning job.

CKZ9DB2D13E Subtask *subtask_number*, Cannot free DD. DD name: *dd_name*, Data set: *data_set_name*, *dynalloc()* error code: *dynalloc_c_function_error_code*, *dynalloc()* info code: *dynalloc_c_function_info_code*.

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ9DB2D14E Subtask *subtask_number*, Unexpected XMAP value (*value*) was encountered while processing OBDREC record at RID *rid*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ9DB2D15E Subtask *subtask_number*, Error while processing OBDREC row at RID *rid*: *error_description*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ9DB2D16E Subtask *subtask_number*, OBDREC fragment at RID *rid* has unexpected record ID. Current/expected record IDs: *current_id/expected_id*

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZ9DB2D17E Subtask *subtask_number*, Spanned record at RID *rid* does not have a suffix

Explanation:

This is an internal error.

User response:

Contact IBM Software Support.

CKZB201I DDL generation module started

Explanation:

This is an informational message specifying that the DDL generation module is running.

User response:

No action is required.

CKZB202I DDL generation module completed with return code = *return_code*

Explanation:

This is an informational message specifying that the DDL generation module completed.

User response:

No action is required.

CKZB203I Module version: *date time*

Explanation:

This is an informational message that shows the version of the module.

User response:

No action is required.

CKZB204I Connected to Db2 subsystem *ssid*

Explanation:

This informational message identifies the Db2 subsystem to which you are connected.

User response:

No action is required.

CKZB205E Error (action *action_value*, code *return_code*, reason *reason_code*) occurred during connection to Db2 subsystem *ssid*

Explanation:

Db2 Cloning Tool cannot connect to the specified Db2 subsystem.

User response:

Review the action, code, and reason information that is included in the message text. Correct the issues and retry connecting.

CKZB206I Disconnected from Db2

Explanation:

This informational message specifies that you have disconnected from the current Db2 subsystem.

User response:

No action is required.

CKZB207E Error (code *return_code*, reason *reason_code*), occurred during disconnection from Db2

Explanation:

An error occurred while Db2 was disconnecting from the specified Db2 subsystem.

User response:

Review the code and reason information that is included in the message text. Correct the issues and retry disconnecting.

**CKZB208W SQL warning occurred in *module_value* on line *line_number*.
SQLCODE = *return_code***

Explanation:

This message warns that the SQL statement in the specified module on the specified line number might not perform as expected. The text of the SQL statement is displayed.

User response:

Review the statement and message to verify the validity of the statement. Processing continues.

**CKZB209E SQL error occurred in *module_value* on line *line_number*.
SQLCODE = *return_code***

Explanation:

An SQL error has occurred in the specified module on the specified line number. The text of the SQL statement is displayed.

User response:

Correct the SQL statement and resubmit.

CKZB210I *SQL_text*

Explanation:

This is an informational message that displays the SQL statement.

User response:

No action is required.

CKZB211E Db2 version *version.module_level* is not supported

Explanation:

The Db2 version that you are attempting to connect to is no longer supported.

User response:

Update the Db2 subsystem to a supported version.

**CKZB212E Specified GENERATE ALTERS FOR XML/LOB value *value* is invalid.
Valid values are Y or N**

Explanation:

An incorrect value was specified for the option GENERATE ALTERS FOR XML/LOB.

User response:

Correct the value.

CKZB213E Cannot open *ddname* for output

Explanation:

The specified DD cannot be opened.

User response:

Verify the existence of the DD and correct the value if needed.

CKZB214E Insufficient memory for allocation

Explanation:

Not enough memory has been allocated for the function that you are performing.

User response:

Allocate additional memory.

CKZB215E Cannot load module *module_name*

Explanation:

Db2 Cloning Tool is unable to load the specified module.

User response:

Verify that the module exists and the name is correct.

CKZB216E The list of input objects is empty

Explanation:

Input list contains no objects.

User response:

Contact IBM Software Support. Provide the contents of the DD CKZDDLDG if possible.

CKZB230E Check existence for *object_key* failed

Explanation:

The object with the specified object key does not exist on the source Db2 subsystem.

User response:

Verify the specified object key.

CKZB231E *object_key* does not exist

Explanation:

The object with the specified object key does not exist on the target Db2 subsystem.

User response:

Verify the specified object key.

CKZB235E Fetch for *object_key* failed

Explanation:

An error occurred during the fetch of object with specified object key.

User response:

Verify the specified object key.

CKZB243I RTS generation module started

Explanation:

This is an informational message specifying that the RTS generation module is running.

User response:

No action is required.

**CKZB244I RTS generation module completed
with return code = *return_code***

Explanation:

This is an informational message specifying that the RTS generation module completed.

User response:

No action is required.

**CKZB245I *table_name* CATALOG UPDATE FOR
object_key SUCCESSFUL**

Explanation:

This is an informational message specifying that the information for the object with specified object key was updated in the specified catalog table.

User response:

No action is required.

CKZB246E Cannot open *ddname* for input

Explanation:

The file in the specified DD statement cannot be opened.

User response:

Verify that the file named in the DD statement is available and correctly specified.

CKZB247E *field* is required

Explanation:

The field that is identified in the message is a required field, but it has not been specified.

User response:

Specify the field.

**CKZB248E Unexpected token was found:
token. Possible tokens are:
expected_tokens.**

Explanation:

An unexpected token that is named in the message is encountered. You must choose a valid token from the list of possible tokens that is provided.

User response:

Specify a valid token value.

**CKZB249E *token control card* is specified
without a value. Value is required**

Explanation:

A value is missing from the specified token control card.

User response:

Specify a value for the token control card.

**CKZB250E *token control card* must be
specified only once**

Explanation:

The specified token control card has been specified more than once.

User response:

Remove all duplicate token control cards.

**CKZB254I *table_name* CATALOG SELECT FOR
object_key SUCCESSFUL**

Explanation:

This informational message indicates that the information about the object with specified object key was selected from the specified catalog table.

User response:

No action is required.

**CKZB256I The value of the sequence
'seqname' increased by value.**

Explanation:

This informational message indicates that the value of the specified sequence was increased by the specified value.

User response:

No action is required.

**CKZB260I Db2 version is *version*,
modification level is *modlevel*.**

Explanation:

This is an informational message identifying the version and modification level of the Db2 subsystem to which you are connected.

User response:

No action is required.

**CKZB266E *token1* and *token2* are equal.
These values must be different**

Explanation:

The values of the specified tokens are equal.

User response:

Correct the values of the tokens.

CKZB267I **Additional processing for XML tables started**

Explanation:

This is an informational message specifying that the additional processing for XML tables is running.

User response:

No action is required.

CKZB268I **Additional processing for XML tables completed with return code = *return_code***

Explanation:

This is an informational message specifying that the additional processing for XML tables has completed.

User response:

No action is required.

CKZB271E ***type key.part* must be created with explicit storage allocation and requires DDL-ATTRIBUTE-CHANGE rule for source VCAT *source_vcat***

Explanation:

A DDL-ATTRIBUTE-CHANGE rule for the specified VCAT does not exist for the specified object.

User response:

Correct the VCAT translation rules in the DDL-ATTRIBUTE-CHANGE section.

CKZB272W ***type key.part* must be created with explicit storage allocation and requires DDL-ATTRIBUTE-CHANGE rule for source VCAT *source_vcat***

Explanation:

A DDL-ATTRIBUTE-CHANGE rule for the specified VCAT does not exist for the specified object.

User response:

Review the message to verify the validity of the statement. Processing continues.

CKZB273E **Table *creator.table_name* specifies DATACAPTURE, but table space *db_name.tablespace_name* does not have LOG specified**

Explanation:

A DDL-ATTRIBUTE-CHANGE rule DATACAPTURE exists for the specified table, but the rule for the LOG attribute is incorrect for the specified table space. If DATACAPTURE is set to CHANGES for the specified table, LOG must be set to YES for the specified table space.

User response:

Correct the LOG and DATACAPTURE translation rules in the DDL-ATTRIBUTE-CHANGE section for the specified objects.

CKZB274E **Invalid BUFFERPOOL Pairing, Source Pool=*src_value*, Target Pool=*trg_value***

Explanation:

A DDL-ATTRIBUTE-CHANGE rule for the specified buffer pools exists, but the buffer pools are not compatible.

User response:

Correct the BUFFERPOOL translation rules in the DDL-ATTRIBUTE-CHANGE section.

CKZB276I **Catalog objects check started**

Explanation:

This is an informational message specifying that the catalog objects check is running.

User response:

No action is required.

CKZB277I **Catalog objects check completed with return code = *return_code***

Explanation:

This is an informational message specifying that the catalog objects check has completed.

User response:

No action is required.

CKZB278E **Catalog objects check failed**

Explanation:

An error occurred during the check process for the existing objects.

User response:

For more detailed information, review the source/target logs.

CKZB279I **Object *object_key* already exists on the Target, DDL will not be generated**

Explanation:

This is an informational message specifying that the object with the specified object key already exists on the target, and DDL for the object will not be generated.

User response:

No action is required.

CKZB280I **The location name of the current server is '*current_server*'**

Explanation:

This is an informational message that specifies the location of the current server.

User response:
No action is required.

CKZB281W ***key_value* has Data Capture Changes turned on. This table may be involved in a replication strategy and may need future corrective action post recover**

Explanation:
The table with the specified object key has the value CHANGES for DATACAPTURE attribute.

User response:
Verify that the table should be present in the cloning scenario. Processing continues.

CKZB288W **The definition of table *creator.table_name* is incomplete, because it lacks required indexes. Constraints for these indexes will be skipped**

Explanation:
The index for the specified table is absent in the cloning scenario. Constraints related to the table will be skipped.

User response:
Verify that the index should be absent from the cloning scenario. Processing continues.

CKZB289I **Db2 SQL level is *sqllevel***

Explanation:
This is an informational message specifying the Db2 SQL level.

User response:
No action is required.

CKZB290W **Translation rules exist for databases with different definitions into one database *objectkey***

Explanation:
The rules for databases with different definitions into one database are present.

User response:
Verify the validity of the database translation rules in the cloning scenario. Processing continues.

CKZB401I **LISTDEF processing module started**

Explanation:
This informational message is from the LISTDEF processing module.

User response:
No action is required.

CKZB402I **LISTDEF processing module completed with return code = *return_code***

Explanation:
This informational message is from the LISTDEF processing module.

User response:
No action is required.

CKZB403I **LISTDEF processing module version: *build_date build_time***

Explanation:
This informational message is from the LISTDEF processing module.

User response:
No action is required.

CKZB404E **Cannot open *ddname* for output**

Explanation:
The output DD *ddname* that is listed in the message cannot be opened.

User response:
If unable to resolve this problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB405E **Insufficient memory for allocation**

Explanation:
The required buffer size cannot be allocated.

User response:
If unable to resolve this problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB406E **Cannot load module *module_name***

Explanation:
The LOAD macro failed to load the indicated module.

User response:
If unable to resolve this problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB407I **Db2 version is *db2_version*, modification level is *db2_mod_level***

Explanation:
This message is informational and shows the version and modification level of the source Db2 subsystem.

User response:
No action is required.

CKZB408I ***record***

Explanation:

This message is informational and is used to print records from DDs.

User response:
No action is required.

CKZB409E DD ddname is already allocated

Explanation:
The ddname that is listed in the message is already allocated. The ddname must not be allocated.

User response:
If unable to resolve this problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB410E DD ddname allocation error

Explanation:
Dynamic allocation for a ddname failed. Processing terminates.

User response:
If unable to resolve this problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB411E DD ddname freeing error

Explanation:
Dynamic deallocation for a ddname failed. Processing terminates.

User response:
If unable to resolve this problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB412W WARN-ON-INCOMPLETE-RI(Y) is specified, but not all LISTDEF statements have RI specified

Explanation:
Not all LISTDEF statements have an RI specification, and WARN-ON-INCOMPLETE-RI parameter is set to Y.

User response:
If RI specification is not required for all LISTDEF input statements, you can safely ignore this message. To avoid this message, set the WARN-ON-INCOMPLETE-RI parameter to N. Otherwise, specify RI for all LISTDEF statements, either manually or by using the INCLUDE-ALL-RI parameter.

CKZB413E Unexpected token was found: token. Possible tokens are: possible_tokens

Explanation:
An error occurred during LISTDEF statement parsing. An unexpected token was encountered.

User response:
Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support.

Have available the full listing that contains this message.

CKZB414E token is specified without a value. A value is required

Explanation:
An error occurred during LISTDEF statement parsing. One of the tokens (CLONED, COPY, LISTDEF) does not have a required value after it (YES/NO or LISTDEF name).

User response:
Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB415E token must be specified only once

Explanation:
An error occurred during LISTDEF statement parsing. One of the tokens was specified more than once in one statement.

User response:
Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB416E Invalid PARTLEVEL value: partlevel_value

Explanation:
An error occurred during LISTDEF statement parsing. The PARTLEVEL value is incorrect.

User response:
Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB417E Type specification is required if object specification is DATABASE or STOGROUP

Explanation:
An error occurred during LISTDEF statement parsing. DATABASE and STOGROUP object specifications require a type specification, which was not found.

User response:
Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB418E No LISTDEF statements were specified

Explanation:
An error occurred during LISTDEF statement parsing. No statements were specified, or only INCLUDE

INDEXSPACES statements were specified with ALLOW-COPY-INDEXSPACES(Y).

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB419E *field is required*

Explanation:

A keyword that is required for processing has been omitted. Processing terminates.

User response:

Specify the required keyword.

CKZB420E *Cannot open ddname for input.*

Explanation:

The input DD *ddname* that is listed in the message cannot be opened.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB421E *Cannot delete module
module_name*

Explanation:

The DELETE macro failed to delete the indicated module.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB422E *Cannot attach module
module_name, code return_code*

Explanation:

The ATTACHX macro failed to attach the indicated module.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB423E *Error has occurred during utility
output parsing*

Explanation:

An internal error occurred during utility output parsing.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB424E *A statement is specified without
an object name. An object name is
required*

Explanation:

An error occurred during LISTDEF statements parsing. A statement is missing the required object name.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB425E *Unexpected end of statements*

Explanation:

An error occurred during LISTDEF statements parsing. A statement is missing the required object specification or type specification.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB426E *PARTLEVEL and RI cannot be both
specified in one statement*

Explanation:

An error occurred during LISTDEF statements parsing. PARTLEVEL and RI options were specified for one statement; both cannot be specified.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB427E *No objects selected by LISTDEF*

Explanation:

The IBM Db2 LISTDEF utility did not return any objects for the specified LISTDEF.

User response:

Check whether the specified LISTDEF rules refer to existing objects. Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB428E *Unexpected end of file ddname*

Explanation:

An unexpected end of file was encountered.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB429E **No LISTDEF statements were returned**

Explanation:

The IBM Db2 LISTDEF utility returned no INCLUDE statements, or an error occurred during utility output parsing.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB430E **Value '%' is not valid for STOGROUP**

Explanation:

An error occurred during LISTDEF statements parsing. The percent sign (%) is not a valid value for a STOGROUP object specification.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB431E **Comment was opened but has not been closed before the end of file**

Explanation:

An error occurred during LISTDEF statements parsing. A comment was opened ('/*'), but was not closed ('*/') before the end of file.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB432I **Input LISTDEF statements are to follow**

Explanation:

This message is informational and is followed by CKZB433I messages that contain input LISTDEF statements without LISTDEF rules applied (ALWAYS-COPY-INDEXSPACES, INCLUDE-ALL-RI, and ALWAYS-COPY-HISTORY-TABLES).

User response:

No action is required.

CKZB433I *listdef_statement*

Explanation:

This message is informational and contains a LISTDEF statement.

User response:

No action is required.

CKZB434I **LISTDEF statements for processing are to follow**

Explanation:

This message is informational and is followed by CKZB433I messages that contain input LISTDEF statements with LISTDEF rules applied (ALWAYS-COPY-INDEXSPACES, INCLUDE-ALL-RI, and ALWAYS-COPY-HISTORY-TABLES).

User response:

No action is required.

CKZB435I **Calling IBM LISTDEF utility**

Explanation:

The LISTDEF processing module is about to call the IBM LISTDEF utility.

User response:

No action is required.

CKZB436E **An error occurred during LISTDEF execution, code = return_code**

Explanation:

An error occurred during IBM LISTDEF utility execution.

User response:

If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB437E **Invalid database name specified, DSNDDB01 | DSNDDB06 | DSNDDB07**

Explanation:

An error occurred during IBM LISTDEF utility execution. An invalid database name was specified. The database names DSNDDB01, DSNDDB06, and DSNDDB07 cannot be specified in LISTDEF.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZB438E **One object qualifier is specified for object specification TABLE/INDEX and DEFAULT-SQLID is not specified**

Explanation:

An error occurred during IBM LISTDEF utility execution. A statement with the object type TABLE or INDEX has specified only the object name, but DEFAULT-SQLID is not provided to set the default creator value.

User response:

Correct the input and resubmit the job. If unable to resolve the problem, contact IBM Software Support. Have available the full listing that contains this message.

CKZDYN01I **SYNTAX ERROR IN VALUE OF
XXXXXX -n | Illegal character in
input text | CKZ01PSN not
available | IEFDB476 not available
| Unknown request type | Function
complete. RC=0 | DSN failed RACF
test | Not enough text units - |
Critical keyword missing: xxxxx |
SVC 99 (F) FAILURE. R15: xx
ERROR: xxxxx INFO: xxxxx**

Explanation:

Indicates CKZ01DYN failed during a dynamic allocation function.

User response:

Report this error and associated product errors to IBM Software Support.

CKZEM000E **Migrate library API is not
initialized**

Explanation:

An internal error occurred during initialization of the migrate library API.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZEM101I **Migration of table space |
subsystem profiles started**

Explanation:

This message marks the beginning of the migration operation for the specified type of cloning profiles.

User response:

No action is required.

CKZEM102I **Table space | Subsystem profile
successfully loaded:
'profile_creator':profile_name'**

Explanation:

The cloning profile that is listed in the message has been successfully loaded.

User response:

No action is required.

CKZEM103I **Table space | Subsystem profile
successfully migrated:
'profile_creator':profile_name'**

Explanation:

The cloning profile that is listed in the message has been successfully migrated.

User response:

No action is required.

CKZEM104I **Migration of table space |
subsystem profiles finished**

Explanation:

This message marks the end of the migration operation for the specified type of cloning profiles.

User response:

No action is required.

CKZEM105I **Simulate mode is enabled. No
changes will be added into the
profile repository**

Explanation:

This message is informational.

User response:

No action is required.

CKZEM106W **Table space | Subsystem profile
has an unsupported version:
'profile_creator':profile_name'**

Explanation:

The cloning profile that is listed in the message has an unsupported version.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZEM107E **The data set is not a profile
repository: HLQ = 'repository_hlq'**

Explanation:

An invalid high-level qualifier was specified for the profile repository.

User response:

Specify a valid profile repository and resubmit the job.

CKZEM108E **An internal error occurred while
working with profile repository:
'system_message'**

Explanation:

An error occurred during processing of the profile repository.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZEM109I **The profile already exists**

Explanation:

There is an existing profile with the same name in the target profile repository.

User response:

No action is required.

CKZEM110I Profile will be skipped

Explanation:

The processed profile will not be migrated because there is an existing cloning profile with the same name.

User response:

No action is required.

CKZEM111I Profile will be replaced

Explanation:

The processed profile already exists in the profile repository. The profile will be overwritten.

User response:

No action is required.

CKZEM112I The new name for the profile will be: 'profile_name'

Explanation:

The processed profile will be imported with a new name because there is an existing cloning profile with the same name.

User response:

No action is required.

CKZEM113I Profile matches to the 'Cloning from Data sets' | 'Cloning from Image copies' | 'Cloning from User copy' scenario

Explanation:

The processed table space cloning profile matches the scenario that is listed in the message.

User response:

No action is required.

CKZEM114W Table space | Subsystem profile migrated with warnings: 'profile_creator': 'profile_name'

Explanation:

The cloning profile that is listed in the message has been migrated. Warnings were encountered during the migration.

User response:

Check previous messages for more information.

CKZEM115W Profile will be skipped. Unsupported version of replaced application | subsystem profile

Explanation:

The processed profile will not be migrated because the cloning profile in the target profile repository has an unsupported version.

User response:

Ensure that the correct and latest versions of all required load libraries are being used.

CKZEM116W Profile will be skipped. No access for replaced application | subsystem profile

Explanation:

The processed profile will not be migrated because the current user does not have permissions to the cloning profile in the target profile repository.

User response:

Either change the permissions for the clone profile in the target repository and try again, or migrate the current profile using MIGRATE-ACTION-IF-PROFILE-EXISTS(NEW).

CKZEM120W Incorrect object name qualifier in listdef: 'object_name'. The new value will be: 'new_object_name'. Fix it manually if needed

Explanation:

This message occurred due to incorrect input data in the source profile.

User response:

If needed, change the value of the described field manually.

CKZEM121W The length of the control HLQ exceeds maximum length (26): 'control_hlq'. Default value will be used. Fix it manually if needed

Explanation:

This message occurred due to incorrect input data in the source profile.

User response:

If needed, change the value of the described field manually.

CKZEM122W Default value will be used for control member. Fix it manually if needed

Explanation:

This message occurred due to incorrect input data in the source profile.

User response:

If needed, change the value of the described field manually.

CKZEM123W Invalid data set and member are specified for the SYSREC | SYSPUNCH | SORTOUT | SYSUT in the Unload-Load command. Default values will be used. Fix it manually if needed

Explanation:

This message occurred due to incorrect input data in the source profile.

User response:

If needed, change the value of the described field manually.

CKZEM124W Invalid data set and member are specified for the PARMLIB. Default values will be used. Fix it manually if needed.

Explanation:

This message occurred due to incorrect values for the data set and member names for the PARMLIB in the CKZINI DD specification of the source profile. After migration, the CKZINI DD will be empty. The ISPF interface will use the data set and member names that are specified in the CKZCLIST CLIST, which is used to run the ISPF interface.

User response:

If needed, change the value of the described field manually.

CKZEM125I Default value will be used for job cards. Fix it manually if needed

Explanation:

The new version of job cards will be used.

User response:

No action is required. If needed, change the value of the described field manually.

CKZEM126E File tailoring API error: 'system_message'

Explanation:

An error occurred during file tailoring processing.

User response:

Ensure that the correct and latest versions of all required load libraries are being used. In addition, ensure that the CKZSLIB DD is allocated. If unable to determine the cause of this error, contact IBM Software Support. Have available the listing that contains this message.

CKZERR00I OBJECT ACCESS FAILURE, DDNAME: ddn

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR01I VVDS ACCESS FAILURE, DDNAME: ddn

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR02I DSN: usercat name

Explanation:

User Catalog cannot be accessed.

User response:

Follow the Programmer Response for message IEC161I found in JESYSMSGs or eliminate the usercat from the selection list.

CKZERR03I NO DD ALLOCATION FOUND

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR04I FAILING module FUNCTION: ##,description

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR05I PREVIOUS module FUNCTION: ##,description

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended Support.

CKZERR06I PROGRAM CSECT: csect

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR08I statement ERROR ID: listingID#

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR09I statement ASM LISTING LINE #: line#

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR10I R15 = ##, description

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR12I module ASM LISTING LINE #: line

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR14I PROCESSOR R15: r15

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR15I PROCESSOR REASON CODE: nnn

Explanation:

This is the decimal value of the failure reason code. Usually this is followed by message CKZERR20I containing text explanation of the reason code. If a text explanation is not available, CKZERR18I is issued, "NO DESCRIPTION FOR REASON CODE".

User response:

If message CKZERR18I follows CKZERR15I and the error is determined by OPEN, CLOSE or VSAM (as indicated by message CKZERR19I), additional information about the reason code can be obtained by consulting IBM manual "z/OS Macro Instruction for Data Sets" section "VSAM Macro Return and Reason Codes". There are sub-sections for "OPEN", "CLOSE" and "Record Management" respectively.

CKZERR16I SVC99 ERROR CODE:

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR17I SVC99 INFORMATION CODE:

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR18I NO DESCRIPTION FOR REASON CODE

Explanation:

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR19I ERROR DETERMINED BY:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR20I description**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR21I ** IMPLICIT OPEN**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR22I ** IMPLICIT CLOSE**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR23I ** IMPLICIT REPOSITION**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR24I OPEN CLASSIFICATION:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR25I CALLING PARM LRECL:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR26I CALLING PARM KEYLEN:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR27I LRECL:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR28I CI RBA REQUESTED:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR29I VVR/NVR KEY REQUESTED:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR30I RECORD TYPE REQUESTED:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZERR31I VVR KEYRANGE QUALIFIER REQUESTED:**Explanation:**

These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response:

Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please contact IBM Software Support.

CKZG1010E The following space is not set to LOG for a required log range.**Explanation:**

The space listed in the messages is not set to LOG for a required log range.

User response:

No action is required.

CKZG1012E No valid full image copy in the SYSCOPY history was found for space(s): spaces**Explanation:**

Db2 Cloning Tool was unable to find a valid full image copy in the SYSCOPY history for the table space(s) indicated in the message. Db2 Cloning Tool requires a full image copy registered in SYSCOPY.

User response:

Ensure the image copy is registered in SYSCOPY and that it is valid.

CKZG1013E A table update ICTYPE was found in SYSCOPY that did not log for space(s): spaces**Explanation:**

There were multiple spaces being processed for which it was impossible for Db2 Cloning Tool to process due to the fact that some operation (such as LOAD REPLACE LOG(NO), REORG LOG(NO), etc.) occurred at some point between the selected starting point and the specified end point.

User response:

No action is required.

CKZG1014I Database: database Space: space Partition: partition**Explanation:**

This message is issued in conjunction with other Db2 Cloning Tool messages to indicate the database, space, and partition for which other messages apply.

User response:

No action is required.

CKZG1015E Could not determine disk/tape status of unit name.**Explanation:**

The device type for work data sets entered in the control file is invalid.

User response:

Enter the correct device type.

CKZG1016E The device type of the unit name from the control file could not be determined.**Explanation:**

The device type for work data sets entered in the control file is invalid.

User response:

Enter the correct device type.

CKZG1017E The REPORT utility returned an unrecoverable error.**Explanation:**

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG1018E The FULL image copy DD CA(LP/LB/RP/RB) {1} is missing from the JCL. Each CAxxxxxx DD correlates to each SPACE(...)
control card group.

Explanation:

The full image copy data set is not included in your Db2 Cloning Tool JCL.

User response:

Verify that the JCL is formatted correctly and contains the necessary information for your Db2 Cloning Tool job.

CKZG1019E **The FULL image copy DD CA {1} refers to a DSNAME already in SYSCOPY.**

Explanation:

You specified a full image copy data set name that already exists in SYSCOPY.

User response:

Specify a different image copy data set name.

CKZG1020I **Each CAxxxx DD correlates to each SPACE(...) control card group.**

Explanation:

Each CAxxxx DD statement must be associated with a corresponding SPACE(...) control card group.

User response:

Verify that the JCL is formatted correctly and that each CAxxxx DD statement is associated with a SPACE(...) control card group.

CKZG1021E **The TO_QUIESCE control card was specified, but no quiesce point was found.**

Explanation:

The TO_QUIESCE control card directs Db2 Cloning Tool to read the log and incorporate data into the image copy up to the most recent quiesce point but no quiesce point was found.

User response:

No action is required.

CKZG1022E **The stop point precedes the start point for space: Database: database Table space: table_space Partition: partition Start point X'startpoint' End point X'endpoint'.**

Explanation:

The Db2 Cloning Tool job will not run if the stop point proceeds the start point for the listed database, table space, partition.

User response:

Correct the JCL and resubmit the job.

CKZG1023I **The version of DB2 subsystem ssid is ver.**

Explanation:

Displays the SSID and the version of the Db2 subsystem.

User response:

No action is required.

CKZG1024I **The version of DB2 group attach member_name is version.**

Explanation:

Displays the version of Db2 group attach that the Db2 group attach member subsystem is running.

User response:

No action is required.

CKZG1025I **Control card stream process complete. Selected space count =space count.**

Explanation:

This is an informational message stating that the control card scanning process has finished. The number of objects found in the control card set is indicated by the *space count* variable.

User response:

No action is required.

CKZG1027I **Db2 Cloning Tool will process dataset for [tablespace | indexspace] space_name.**

Explanation:

Indicates the data set name that Db2 Cloning Tool will process.

User response:

No action is required.

CKZG1028I **The image copy is of all parts.**

Explanation:

Indicates that the image copy is of all partitions of the table space.

User response:

No action is required.

CKZG1029I **The image copy contains one partition (partition).**

Explanation:

Indicates the one partition that the image copy contains.

User response:

No action is required.

CKZG1030E **A concurrent image copy was found in the SYSCOPY history. It cannot be used.**

Explanation:

The DFDSS concurrent image copy that was found cannot be read by Db2 Cloning Tool.

User response:

Select an alternative mechanism by which to recover the space.

CKZG1031I Only partition *partition* within the image copy will be updated with log data and written to an individual partition copy.

Explanation:

Db2 Cloning Tool will only update the partition within the image copy with log data and will write to an individual partition image copy.

User response:

No action is required.

CKZG1032I All partitions will be updated with log data.

Explanation:

Db2 Cloning Tool will update all partitions with log data.

User response:

No action is required.

CKZG1033E The specified dataset name does not exist.

Explanation:

The data set specified for the TO_IC parameter does not exist.

User response:

Specify the correct data set name for the TO_IC parameter.

CKZG1034I Db2 Cloning Tool will process the log only for [table space | index space] *space_name PART part*.

Explanation:

Db2 Cloning Tool will process only the log for the indicated table space or index space and partition.

User response:

No action is required.

CKZG1035E An image copy was found, but its RBA precedes the logging start point.

Explanation:

This message indicates that although an image copy was found, it could not be used since its RBA precedes the logging start point.

User response:

No action is required.

CKZG1036I Db2 Cloning Tool processing ends.

Explanation:

Indicates that Db2 Cloning Tool processing has completed.

User response:

No action is required.

CKZG1038E An incremental image copy was marked as cataloged in SYSCOPY, but was not found in the MVS catalog.

Explanation:

This message indicates that although an incremental image copy was marked as cataloged in SYSCOPY, it was not found in the MVS catalog.

User response:

No action is required.

CKZG1039E At least two end points within a single GROUP() are not the same.

Explanation:

At least two end points within a GROUP are not the same.

User response:

Verify that the end points you defined are correct.

CKZG1040I The SPACE(...) set involved that the error was detected in was #'XXXXX'

Explanation:

There was an error in the SPACE set indicated in the message.

User response:

Verify the correct SPACE syntax has been specified.

CKZG1155I Control card stream processed by Db2 Cloning Tool follows...

Explanation:

Indicates the control card stream that was processed by Db2 Cloning Tool.

User response:

No action is required.

CKZG1156I Db2 Cloning Tool processing messages follow...

Explanation:

Indicates that there are Db2 Cloning Tool messages that follow.

User response:

Evaluate the message as necessary.

CKZG1200E The subsystem Db2 Cloning Tool was started with could not be found in JES2.

Explanation:

The subsystem Db2 Cloning Tool was started with could not be found in JES2.

User response:

Verify that you have specified the correct subsystem.

CKZG1201E **The subsystem Db2 Cloning Tool was started with is not active in JES2.**

Explanation:

This message indicates that the subsystem that Db2 Cloning Tool was started with is not active in JES2.

User response:

No action is required.

CKZG1202E **There are no active DB2 members on this machine for this data sharing group.**

Explanation:

The data sharing group you specified does not have any active Db2 members so Db2 Cloning Tool processing cannot proceed.

User response:

Specify a valid data sharing group attach name or a valid subsystem on which the Db2 Cloning Tool processing can run.

CKZG1203I **DB2 subsystem is not defined to OS/390. Using group attach name instead.**

Explanation:

The Db2 subsystem you specified is not defined on OS/390®. If you use a group attach name, you will be able to connect to a Db2 subsystem that is active on OS/390.

User response:

Edit your Db2 Cloning Tool setup to connect to a group attach name or to connect to a Db2 subsystem that is active on OS/390.

CKZG1204I **DB2 subsystem is not active on OS/390. Using group attach name instead.**

Explanation:

The Db2 subsystem you specified is not active on OS/390. If you use a group attach name, you will be able to connect to a Db2 subsystem that is active on OS/390.

User response:

Edit your Db2 Cloning Tool setup to connect to a group attach name or to connect to a Db2 subsystem that is active on OS/390.

CKZG1205I **The subsystem Db2 Cloning Tool was started with is the group attach name.**

Explanation:

This message indicates the subsystem group attach name that Db2 Cloning Tool process is using.

User response:

No action is required.

CKZG1206I **The following subsystems are part of the data sharing group.**

Explanation

This message, in conjunction with message CKZG1207I, provides the following information about the subsystem on which your Db2 Cloning Tool job ran:

- Subsystem—the subsystem.
- Member ID—the member ID.
- Defined to OS/390—whether this member is defined to OS/390.
- Active—whether this member is known to this OS/390 running on OS/390.

Note: Db2 Cloning Tool cannot detect the status of a member that is not running on this OS/390. Although a Db2 member may appear to be inactive, it may be running on another OS/390. Regardless, Db2 Cloning Tool reads the logs and processes all of the necessary files from each member of the data sharing group.

User response:

No action is required.

CKZG1207I **Subsystem: *subsystem* Member ID: *memberid* Defined to OS/390: *system* Active: *status***

Explanation

This message, in conjunction with message CKZG1206I, provides the following information about the subsystem on which your Db2 Cloning Tool job ran:

- The subsystem.
- The member ID.
- Whether or not this member is defined to OS/390.
- Whether or not this member is running on OS/390.

Note: Db2 Cloning Tool cannot detect the status of a member that is not running on this OS/390. Although a Db2 member may appear to be inactive, it may be running on another OS/390. Regardless, Db2 Cloning Tool reads the logs and processes all of the necessary files from each member of the data sharing group.

User response:

No action is required.

CKZG1208I ***ssids***

Explanation:

This message displays the SSIDs that accompany messages CKZG1206I and CKZG1207I.

User response:

No action is required.

CKZG1300I **The ENQs for the spaces were successful.**

Explanation:

This message indicates that the ENQs for the table spaces completed successfully.

User response:

No action is required.

CKZG1301E **The ENQ for database *database* PART *part* was not successful.**

Explanation:

Indicates the database and partition for which the ENQs did not complete successfully.

User response:

No action is required.

CKZG1402E **Could not start log record writer.**

Explanation:

A z/OS attachment error occurred attempting to start a component of the load read phase process.

User response:

Ensure the product library is complete. Contact IBM Software Support.

CKZG1416E **A FTR sort program could not be started.**

Explanation:

The SORT program could not be started.

User response:

Contact IBM Software Support.

CKZG1417E **An invalid return code was detected from the SORT program (FTR).**

Explanation:

The SORT program ended with an error.

User response:

Contact IBM Software Support.

CKZG1418I **All start points are Sharelevel Reference; checkpoint processing skipped.**

Explanation:

This informational message indicates that checkpoint processing has been skipped since all start points are Sharelevel Reference.

User response:

No action is required.

CKZG1419E **A mismatch between passed Zparm information and the JES SSCT was found.**

Explanation:

The ZPARM member for the source subsystem could not be found.

User response:

Verify that the ZPARM information is accurate in the log apply job. Verify that the correct data sets containing the ZPARM member for the source subsystem are allocated. Contact IBM Software Support if the problem persists.

CKZG1420E **While trying to read *zparm_name* information, program finished with code *hexadecimal_return_code*.**

Explanation

This message is displayed if an unexpected error occurred. The message displays the error return code when the program is trying to provide information about Db2 ZPARM fields. Description of error codes:

- 00008: Open for Db2 libraries failed.
- 00012: Load for ZPARM module failed.
- 00116: The first Db2 LOADLIB in the subsystem concatenation is blank. Check control file setup; if everything is correct, contact IBM Software Support.
- 00120: One of the Db2 LOADLIBs in the concatenation could not be allocated. Check control file setup; if everything is correct, contact IBM Software Support.
- 00124: The Db2 LOADLIBs in the concatenation list could not be concatenated. Check control file setup; if everything is correct, contact IBM Software Support.
- 00208: At least one of the token names was not found in the token list. Check control file setup; if everything is correct, contact IBM Software Support.
- 00212: The control file could not be opened. Check control file setup; if everything is correct, contact IBM Software Support.

User response:

Check control file setup if the description of the error code requires it; if not, or if the control file is correct, contact IBM Software Support.

CKZG1421E **An unexpected error occurred while trying to read the bootstrap data set.**

Explanation:

The ZPARM member could not be found.

User response:

Verify that the ZPARM information is accurate in the log apply job. Verify that the correct data sets containing the ZPARM member are allocated. Contact IBM Software Support if the problem persists.

CKZG1503I ***task_number data_set_name***

Explanation:

This message accompanies CKZG1502E and indicates the log data set name that was last usable.

User response:

No action is required.

CKZG1510I *task_number* Error code #1: 'code'
#2: 'code'

Explanation:

This message indicates an internal error that occurs when the log reader process cannot allocate an active or archive log file

User response:

Diagnose the problem using the return codes listed in the message. Refer to <https://www.ibm.com/support/knowledgecenter/en/SSEPEK> for information about Db2 messages and codes.

CKZG1515I *task_number* Log accumulated processing is beginning on subsystem *ssid*.

Explanation:

Log accumulated processing has started on the subsystem indicated in the message.

User response:

No action is required.

CKZG1516E *task_number* Above the bar storage exhausted.

Explanation:

The above the bar storage has been exhausted due to system or control card limits.

User response:

The limit set by the maximum secondary allocation parameter has been met.

CKZG1518I *task_number* Reading type log log.

Explanation:

The ACTIVE/ARCHIVE log *log_dataset_name* is about to be read for log records associated with the specified objects.

User response:

No action is required.

CKZG1520I *task_number* The log reader task #*task_number* finished.

Explanation:

This message indicates that the processing of the log reader has completed.

User response:

No action is required.

CKZG1521I Issuing HRECALL for log dataset *dsname*

Explanation:

This is an informational message indicating that the log data set needed for processing has been recalled from migration.

User response:

No action is required.

CKZG1606I The DB2 log record sort DD *ddname* was allocated.

Explanation:

This message displays the input DDNAME. This message is output if Db2 Cloning Tool dynamically allocates the SORTIN2/SORTOUT2 DDNAMES. If the SORTIN2/SORTOUT2 DDNAMES are specified by the user in the step JCL, the messages does not appear and those DDs will be used as specified.

User response:

No action is required.

CKZG1607I The DB2 log record sort DD *ddname* was allocated.

Explanation:

This message displays the output DDNAME. This message is output if Db2 Cloning Tool dynamically allocates the SORTIN2/SORTOUT2 DDNAMES. If the SORTIN2/SORTOUT2 DDNAMES are specified by the user in the step JCL, the messages does not appear and those DDs will be used as specified.

User response:

No action is required.

CKZG1805I Dynamic allocation return code=*rc*.

Explanation:

This message indicates the dynamic allocation return code.

User response:

No action is required.

CKZG1806I Image copy name=*image_copy_name* RBA=*rba*

Explanation:

Indicates the image copy name and RBA.

User response:

No action is required.

CKZG1808I The full image copy *image_copy* could not be deallocated.

Explanation:

Db2 Cloning Tool could not deallocate the full image copy.

User response:

Verify that the file is not in use or damaged. Check with your systems administrator to ensure you have proper authorizations to access this file.

CKZG1825I **The \$ORTPARM override DD is present in the job step JCL.**

Explanation:

The //\$ORTPARM DD was found in the job step JCL. Db2 Cloning Tool will not attempt to allocate and load the parameter file.

User response:

No action is required.

CKZG1826I **The \$ORTPARM DD could not be loaded. Using system defaults.**

Explanation:

An attempt to create and load the //\$ORTPARM file was not successful. Installation defaults will be used instead.

User response:

No action is required.

CKZG1827I **The \$ORTPARM DD could not be allocated. Using system defaults.**

Explanation:

An attempt to create and load the //\$ORTPARM file was not successful. Installation defaults will be used instead.

User response:

No action is required.

CKZG1828I **The \$DB2PRM\$ override DD is present in the job step JCL.**

Explanation:

The //\$B2PRM\$ DD was found in the job step JCL. Db2 Cloning Tool will not attempt to allocate and load the parameter file.

User response:

No action is required.

CKZG1829I **The \$DB2PRM\$ DD could not be loaded. Using system defaults.**

Explanation:

An attempt to create and load the //\$B2PRM\$ file was not successful. Installation defaults will be used instead.

User response:

No action is required.

CKZG1830I **The \$DB2PRM\$ DD could not be allocated. Using system defaults.**

Explanation:

An attempt to create and load the //\$DB2PRM\$ file was not successful. Installation defaults will be used instead.

User response:

No action is required.

CKZG1900I ***task_number* Log range LRSN X'lrns' to X'lrns' is being processed.**

Explanation:

Indicates the log range that is being processed by Db2 Cloning Tool.

User response:

No action is required.

CKZG1901I ***task_number* Log range RBA X'rba' to X'rba' is being processed.**

Explanation:

Indicates the log range that is being processed by Db2 Cloning Tool.

User response:

No action is required.

CKZG2000E **The output full image copy *image_copy* could not be opened.**

Explanation:

Db2 Cloning Tool could not open the output full image copy.

User response:

Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2001E **Dynaloc function error DSN *DSN* rc= *rc* reason= *reason*.**

Explanation:

A call to z/OS dynamic allocation failed.

User response:

Contact IBM Software Support.

CKZG2002E **Error process IDCAM output. Output follows: *output***

Explanation:

IDCAMS system service request returned an error condition. The IDCAMS output and error messages follow.

User response:

Verify IDCAMS output. Contact IBM Software Support.

CKZG2003E **An internal error occurred in the Db2 Cloning Tool merge section.**

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG2005E **The number of pages in the incr. image copy is inconsistent with the page size.**

Explanation:

The page size you specified is not consistent with the number of pages in the incremental image copy.

User response:

Correct the number of pages specified and resubmit the job.

CKZG2006E **An unexpected EOF was encountered on the sorted log records file.**

Explanation:

An unexpected end of file was encountered.

User response:

Contact IBM Software Support.

CKZG2007E **The selected full IC has a DBID/PSID mismatch to the DB2 catalog.**

Explanation:

The DBID/PSID for the selected full image copy does not match those in the Db2 catalog.

User response:

Correct the DBID/PSID for the selected full image copy.

CKZG2008I **Number of pages read from the full image copy file(s)=n.**

Explanation:

This informational message indicate the number of pages that were read from the full image copy files.

User response:

No action is required.

CKZG2009I **Number of pages read from the incremental image copy file(s)=n.**

Explanation:

This informational message indicates the number of pages that were read from the incremental image copy data set(s).

User response:

No action is required.

CKZG2010I **Number of records read from the log apply file=n.**

Explanation:

This informational message indicates the number of pages that were read from the log apply file.

User response:

No action is required.

CKZG2011I **Number of pages written to the new full image copy file(s)=n.**

Explanation:

This informational message indicates the number of pages that were written to the new full image copy data set(s).

User response:

No action is required.

CKZG2012I **Number of pages written to the table/index space file(s)=n.**

Explanation:

Indicates the number of pages written to the table/index space files.

User response:

No action is required.

CKZG2013I **Since no changes were found for this data set, it has been deleted: dsn**

Explanation:

This message appears during dynamic allocation of an output image copy data set and no output was written to that data set. It is similar to the other message that is reported when no output is written to an output image copy data set in JCL.

User response:

No action is required.

CKZG2014E **Error to start rebuild indexes thread.**

Explanation:

An attempt to start the rebuild indexes process returned an error.

User response:

Verify STEPLIB contents for complete install. Contact IBM Software Support.

CKZG2015E **A open failure occurred on the VSAM I/O module.**

Explanation:

An open failure occurred for the VSAM I/O module.

User response:

Refer to message CKZG2023E for any dynamic allocation return codes and consult with your systems programmer. Refer to the *z/OS MVS Programming: Authorized Assembler Service Guide (SA23-1371)* for more information.

CKZG2016E **A close failure occurred on the VSAM I/O module.**

Explanation:

A close failure occurred for the VSAM I/O module.

User response:

Contact IBM Software Support.

CKZG2017E **A write failure occurred on the VSAM I/O module.**

Explanation:

A write failure occurred for the VSAM I/O module.

User response:

Contact IBM Software Support.

CKZG2018E **An open for update failure occurred on the VSAM I/O module.**

Explanation:

An open failure occurred for the VSAM I/O module.

User response:

Contact IBM Software Support.

CKZG2019E **A random fetch failure occurred on the VSAM I/O module.**

Explanation:

A fetch failure occurred for the VSAM I/O module.

User response:

Contact IBM Software Support.

CKZG2020E **A random write failure occurred on the VSAM I/O module.**

Explanation:

A write failure occurred for the VSAM I/O module.

User response:

Contact IBM Software Support.

CKZG2021E **A random close failure occurred on the VSAM I/O module.**

Explanation:

A close failure occurred for the VSAM I/O module.

User response:

Contact IBM Software Support.

CKZG2022E **The underlying table/index space data set could not be found in MVS.**

Explanation:

The table/index space could not be found in MVS.

User response:

No action is required.

CKZG2023E **Dynamic allocation return code = 'return_code'.**

Explanation:

This diagnostic message indicates data set allocation failure.

User response:

Diagnose the problem using the return code. Refer to IBM Knowledge Center for information about Db2 messages and codes.

CKZG2024I **Object Database=*database* Space Name=*space_name* Partition=*partition* will have an image copy written anyway due to control card FORCE_COPIES.**

Explanation:

Db2 Cloning Tool will write an image copy for the object indicated in the message and override the WRITE_TO_VSAM control card, because the control card FORCE_COPIES has been specified with a value of Y.

User response:

No action is required. If you do not want an image copy produced, specify FORCE_COPIES N.

CKZG2025E **The SYSIN DD could not be allocated during WRITE_TO_VSAM processing.**

Explanation:

Db2 Cloning Tool was unable to allocate the SYSIN DD during WRITE_TO_VSAM processing.

User response:

No action is required.

CKZG2026E **A CELL64 free request failed.**

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG2027E **Rebuild indexes thread returned error.**

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG2028E **Log apply process cancelled by request from task manager.**

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG2029I **Space *database.spacename* Part # *number* will be written to DSN.**

Explanation:

SWITCH_VCAT keyword in effect, data set name *dsn* was generated to place WRITE_TO_VSAM result.

User response:

None.

CKZG2030E **Data set organization is not VSAM DSN *DSN*.**

Explanation:

A WRITE_TO_VSAM operation to a specified data set with the REUSE option could not be completed because the data set is not VSAM.

User response:

Check the data sets that were involved. Contact IBM Software Support.

CKZG2031E The LP image copy spanned tape could not be freed for a device switch.

Explanation:

The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response:

Verify that the spanned tape is available for allocation.

CKZG2032E The initial LP image copy could not be allocated on the tape device.

Explanation:

The allocation of the image copy data set to the tape device failed.

User response:

Verify that the tape device is available for allocation.

CKZG2033E The LP image copy data set to be created on tape could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2034E The LP image copy could not be allocated to the DASD device.

Explanation:

The dynamic allocation of the data set to the DASD device failed.

User response:

Verify that the device name is correct and that it is available for allocation.

CKZG2035E The LP image copy data set to be created on DASD could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2036E The spanned LP image copy on tape could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2037E The LB image copy spanned tape could not be freed for a device switch.

Explanation:

The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response:

Verify that the spanned tape is available for allocation.

CKZG2038E The initial LB image copy could not be allocated onto the tape device.

Explanation:

The allocation of the image copy data set to the tape device failed.

User response:

Verify that the tape device is available for allocation.

CKZG2039E The LB image copy data set to be created on tape could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2040E The LB image copy could not be allocated to the DASD device.

Explanation:

The dynamic allocation of the data set to the DASD device failed.

User response:

Verify that the device name is correct and that it is available for allocation.

CKZG2041E The LB image copy data set to be created on DASD could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2042E The spanned LB image copy on tape could not be opened.**Explanation:**

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2043E The RP image copy spanned tape could not be freed for a device switch.**Explanation:**

The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response:

Verify that the spanned tape is available for allocation.

CKZG2044E The initial RP image copy could not be allocated onto the tape device.**Explanation:**

The allocation of the image copy data set to the tape device failed.

User response:

Verify that the tape device is available for allocation.

CKZG2045E The RP image copy data set to be created on tape could not be opened.**Explanation:**

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2046E The RP image copy could not be allocated to the DASD device.**Explanation:**

The dynamic allocation of the data set to the DASD device failed.

User response:

Verify that the device name is correct and that it is available for allocation.

CKZG2047E The RP image copy data set to be created on DASD could not be opened.**Explanation:**

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2048E The spanned RP image copy on tape could not be opened.**Explanation:**

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2049E The RB image copy spanned tape could not be freed for a device switch.**Explanation:**

The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response:

Verify that the spanned tape is available for allocation.

CKZG2050E The initial RB image copy could not be allocated onto the tape device.**Explanation:**

The allocation of the image copy data set to the tape device failed.

User response:

Verify that the tape device is available for allocation.

CKZG2051E The RB image copy data set to be created on tape could not be opened.**Explanation:**

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2052E The RB image copy could not be allocated to the DASD device.**Explanation:**

The dynamic allocation of the data set to the DASD device failed.

User response:

Verify that the device name is correct and that it is available for allocation.

CKZG2053E The RB image copy data set to be created on DASD could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2054E The spanned RB image copy on tape could not be opened.

Explanation:

The image copy data set that is to be created cannot be opened.

User response:

Verify that the image copy data set you specified in your Db2 Cloning Tool JCL is available for use and resubmit the Db2 Cloning Tool job.

CKZG2055I A volume written to and left on the system could not be found.

Explanation:

When Db2 Cloning Tool finishes writing to a tape data set, the tape cartridge is not rewound and ejected. It is left on the tape drive in case another data set needs to be written afterwards. Once any one data set is written, it is closed and code then goes back and reads internal MVS control blocks to get specifics about that data set. If this subsequent code cannot find the data set just written and closed, the error occurs.

User response:

Contact IBM Software Support.

CKZG2057E The following mini log data set could not be deallocated from OS/390:

Explanation:

The mini log data set could not be deallocated from OS/390 and could therefore not be used in Db2 Cloning Tool processing. This message is followed by CKZG2060I which displays the name of the mini log data set that could not be deallocated.

User response:

Verify that the mini log data set is available for use.

CKZG2058E The following mini log data set could not be opened:

Explanation:

The mini log data set could not be opened and could therefore not be used in CKZG processing. This message is followed by CKZG2060I which displays the name of the mini log data set that could not be opened.

User response:

Verify that the mini log data set is available for use.

CKZG2059E The following mini log data set could not be allocated:

Explanation:

The mini log data set could not be allocated and could therefore not be used in Db2 Cloning Tool processing. This message is followed by CKZG2060I which displays the name of the mini log data set that could not be allocated.

User response:

Verify that the mini log data set is available for use.

CKZG2060E *dsname*

Explanation:

This message displays a data set name that is associated with other messages.

User response:

No action is required.

CKZG2061I The mini log file *filename* has been processed.

Explanation:

The mini log indicated in the messages has been processed.

User response:

No action is required.

CKZG2062I CKZG will attempt to use the MINI_LOG_DSN_2 data set instead.

Explanation:

Db2 Cloning Tool was unable to use the MINI_LOG_DSN_1 data set so it will now attempt to use the MINI_LOG_DSN_2 data set.

User response:

No action is required.

CKZG2063E A corrupted row was found in the mini log control table.

Explanation:

Db2 Cloning Tool was unable to use the mini log control table due to a corrupted row.

User response:

No action is required.

CKZG2064E Both mini log data sets for this space could not be opened.

Explanation:

Db2 Cloning Tool attempted to open both mini log data sets for the space but was unable to do so.

User response:

Verify that the mini log data sets are available for use.

CKZG2065I Number of records read from the merged mini log file(s)=

Explanation:

The number of records displayed in this message were read from the merged mini log files.

User response:

No action is required.

CKZG2066E **An unexpected EOF was encountered on a merged mini log records file.**

Explanation:

Db2 Cloning Tool encountered an unexpected EOF on a merged mini log record file.

User response:

No action is required.

CKZG2067E **XLAT_DSN <DSN> will be used for <XLAT_TARGET>**

Explanation:

XLAT_DSN <DSN> was not found and there is no XLAT_VSAM or XLAT_COPY specified. XLAT_TARGET will be determined by format of DSN. So if DSN conforms to Db2 space name format, new VSAM data set will be allocated, if not new sequential data set will be allocated.

User response:

No action is required.

CKZG2068E **The XML sequence number update process failed.**

Explanation:

Coordinating the internal XML sequence number during OBIDXLAT processing could not be completed.

User response:

Contact IBM Software Support.

CKZG2069E **The space *space* resulted in the error condition.**

Explanation:

This message follows other error messages and identifies the space related to the errors.

User response:

No action is required.

CKZG2069I **The space *space* resulted in the error condition.**

Explanation:

Generic message that follows many other error messages.

User response:

No action is required.

CKZG2070E **The alternative SSID XML sequence column update program failed.**

Explanation:

Coordinating the internal XML sequence number during OBIDXLAT processing could not be completed.

User response:

Contact IBM Software Support.

CKZG2071E **An XML update job is needed, but the XML output DSN is missing.**

Explanation:

The XML output DSN was not specified.

User response:

Specify an XML output DSN.

CKZG2072E **An XML update job is needed, but the XML output prefix is missing.**

Explanation:

The XML output prefix was not specified.

User response:

Specify the XML output prefix.

CKZG2073E **An XML update job is needed, but the XML template DSN is missing.**

Explanation:

The XML template DSN was not specified.

User response:

Specify an XML template DSN.

CKZG4504E **The XML template data set could not be allocated.**

Explanation:

Unable to allocate the needed DSN.

User response:

Make sure the DSN exists and is accessible.

CKZG2075E **Control file loadlib information could not be obtained for *ssid***

Explanation:

The control file is not up to date with this Db2 SSID.

User response:

Update it via setup option 0.

CKZG2076E **The XML template data set could not be opened.**

Explanation:

The data set was allocated but could not be opened.

User response:

Contact IBM Software Support.

CKZG2077E **The XML job output data set/member could not be allocated.**

Explanation:

The supplied data set could not be allocated.

User response:

Make sure authority exists to allocate.

CKZG4508E **The XML job output data set/
member could not be opened.**

Explanation:

The data set was allocated but could not be opened.

User response:

Check for proper access authority.

CKZG2079E **The XML template does not
conform to the automatically
generated guidelines.**

Explanation:

The XML template generated by Db2 Cloning Tool has been altered to the point that it does not conform to expected design.

User response:

Regenerate the XML template.

CKZG2080E **The target SSID for XML
translation is missing in the
control cards.**

Explanation:

There is a missing parameter.

User response:

Correct the JCL and resubmit the job.

CKZG2081I **The SPACE(...) set involved that
the error was detected in was
*spacesetnumber***

Explanation:

Generic message that follows many other messages.

User response:

No action is required.

CKZG2082E **The XML target SSID/DBname/
TSname control cards are missing.**

Explanation:

Missing control cards in the Space(...) set.

User response:

No action is required.

CKZG2083E **The XML target SSID/DBname/
TSname control cards are invalid.**

Explanation:

Syntax error in control cards.

User response:

Correct the syntax.

CKZG2084I **XML update job created for
SSID='ssid'.**

Explanation:

The job has been created.

User response:

No action is required.

CKZG2085E **No references to subsystem could
not be found in the JES SSCT.**

Explanation:

The specified Db2 SSID is not defined to z/OS.

User response:

Ensure that the name is correct or contact IBM Software Support.

CKZG2086E **The sorted log file could not be
allocated.**

Explanation:

An allocation error has occurred.

User response:

Verify that the proper authorization is set.

CKZG2087E **The sorted log file could not be
opened.**

Explanation:

After allocating, could not open.

User response:

Ensure proper authorization exists, or contact IBM Software Support.

CKZG2088E **A log record page number
exceeded the extent size
boundary.**

Explanation:

A Db2 internal error occurred. The page number encoded into the log record points beyond the number of allowable pages for a Db2 extent.

User response:

Send the dump and any table space / table creation details to IBM Software Support.

CKZG2091E **The mini log data set
minilog_dsname could not be
allocated.**

Explanation:

The mini log data set could not be allocated and therefore could not be used in Db2 Cloning Tool processing.

User response:

Verify that the mini log data set is available for use.

CKZG2092E **The mini log data set
minilog_dsname could not be
opened.**

Explanation:

The mini log data set could not be opened and therefore could not be used in Db2 Cloning Tool processing.

User response:

Verify that the mini log data set is available for use.

CKZG2095I The sort of the applicable log records was successful.

Explanation:

This message indicates that the sort of the applicable log records completed without error.

User response:

No action is required.

CKZG2098E An invalid return code was detected from the SORT program.

Explanation:

This attempt to SORT returned an error.

User response:

Verify SORT program error messages. Contact IBM Software Support.

**CKZG2099E Invalid image copy DSN:
image_copy_dsname**

Explanation:

This message indicates that an error occurred when trying to process the image copy.

User response:

Verify that the image copy specified in the message is available for use. If the problem persists, contact IBM Software Support.

CKZG2300E An internal error occurred unloading a mini log data set.

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG2301E The following mini log data set could not be deallocated from OS/390:

Explanation:

The specified mini log data set could not be deallocated from OS/390.

User response:

Verify that you have specified the correct mini log data set name generation string.

CKZG2302E The following mini log data set could not be opened:

Explanation:

The specified mini log data set could not be opened.

User response:

Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2303E The following mini log data set could not be allocated:

Explanation:

The specified mini log data set could not be allocated.

User response:

Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2304I dsn

Explanation:

Indicates the mini log DSN. This message is issued in conjunction with message CKZG2303I.

User response:

No action is required.

CKZG2305I Dynamic allocation return code = 'return_code'

Explanation:

This diagnostic message indicates data set allocation failure.

User response:

Diagnose the problem using the return code. Refer to <https://www.ibm.com/support/knowledgecenter/en/SSEPEK> for information about Db2 messages and codes.

CKZG2310I The mini log file: mini_log_file has been processed.

Explanation:

This message indicates the mini log file that has been processed.

User response:

No action is required.

CKZG2312E A space level mini log DSN has the same name as a group level mini log DSN.

Explanation:

Different groups of spaces in the log apply control card have the same mini log data set specified.

User response:

Adjust the naming in the control cards.

CKZG2313E Mini log data set dsn could not be appended because a gap is found for the object in the mini log control table.

Explanation:

There is a gap for the object in the mini log chain in the mini log control table. For this reason, the mini log data set indicated in the message could not be appended.

User response:

To resolve this issue, either remove the mini log DSN from the mini log control table and MVS catalog or specify a new DSN for the mini log.

CKZG2401E The space space PART part has an unknown space status.

Explanation:

This message ensures that the indicated space is to be stopped before proceeding with the WRITE_TO_VSAM process. Db2 Cloning Tool checks the space with a call similar to a '-display db(dbname) spacename(tsname) part(0)' to verify that the space is in 'stop' status. This message displays when the space comes back with a status not equal to RO, STOP, RW, or UT.

User response:

Stop the indicated space before attempting to proceed with the WRITE_TO_VSAM process.

CKZG2402E The stop status check for space space PART part timed out.

Explanation:

This message is output when Db2 Cloning Tool tries to start and it has to ensure that when doing WRITE_TO_VSAM processing that the spaces are indeed stopped. The stop step that is generated (prior to Db2 Cloning Tool) to do this sends commands to Db2 to stop the data sets, but it does not wait for the spaces to actually stop. If an in-flight URID is processing against the object and the stop is done, the space changes to 'STOPP' or stop pending until the URID finishes. It may also take Db2 some time to flush buffers. In either case, Db2 Cloning Tool does a check on the spaces before doing any real processing. If any of the spaces do not come back 'stop,' it waits a few seconds and checks again. After a few checks like this, it aborts, producing this message.

User response:

Diagnose why the space will not stop.

CKZG2500E Fetching SYSIBM.SYSLOGRANGE data produced an error

Explanation:

Db2 Cloning Tool encountered an error when attempting to fetch SYSIBM.SYSLOGRANGE data.

User response:

No action is required. The report utility's output will be output after this message.

CKZG2501E REPORT utility text follows: text

Explanation:

This message is the header line for the REPORT utility output that follows on the next line.

User response:

No action is required.

CKZG2502I Skipping SYSIBM.SYSLGRNX processing.

Explanation:

This informational messages indicates that Db2 Cloning Tool is not processing SYSIBM.SYSLGRNX because NO_SYSLGRNX was specified.

User response:

No action is required.

CKZG2609I The LOG_COPY_PREFERENCE parameter was specified, but no value was found with it.

Explanation:

Your JCL includes the LOG_COPY_PREFERENCE parameter but no value is specified with it.

User response:

Specify a valid value for the LOG_COPY_PREFERENCE parameter.

CKZG2670I The MINILOG_SHARELEVEL keyword is ignored when not producing mini logs.

Explanation:

You included the MINILOG_SHARELEVEL keyword in your JCL indicating the type of SHARELEVEL for mini logs but did not specify the production of producing mini logs. The MINILOG_SHARELEVEL keyword is therefore ignored.

User response:

No action is required.

CKZG2672W The REPAIR_RECOVER_PENDING keyword is ignored when only writing to copies.

Explanation:

The REPAIR_RECOVER_PENDING keyword is specified but this parameter is ignored when writing to copies.

User response:

REPAIR_RECOVER_PENDING is ignored when writing to copies so the REPAIR_RECOVER_PENDING control card can be removed.

CKZG2703I Image copy name=image_copy_name RBA='rba'.

Explanation:

Indicates the image copy name an RBA.

User response:

No action is required.

CKZG2706I Db2 Cloning Tool will process the following incremental image copy file(s):

Explanation:

Db2 Cloning Tool will process the incremental image copy file(s) listed in this message.

User response:

No action is required.

CKZG2707I For table space: table_space PART part

Explanation:

This message indicates the table space and partition related to other Db2 Cloning Tool messages that have been issued.

User response:

No action is required.

CKZG2804E **An unexpected error occurred while trying to read the bootstrap data set.**

Explanation:

An unexpected error was encountered.

User response:

Contact IBM Software Support.

CKZG2805E **While trying to read *zparm_name* information, program finished with code *hexadecimal_return_code*.**

Explanation

This message is displayed if an unexpected error occurred. The message displays the error return code when the program is trying to provide information about Db2 ZPARM fields. Description of error codes:

- 00008: Open for Db2 libraries failed.
- 00012: Load for ZPARM module failed.
- 00116: The first Db2 LOADLIB in the subsystem concatenation is blank. Check control file setup; if everything is correct, contact IBM Software Support.
- 00120: One of the Db2 LOADLIBs in the concatenation could not be allocated. Check control file setup; if everything is correct, contact IBM Software Support.
- 00124: The Db2 LOADLIBs in the concatenation list could not be concatenated. Check control file setup; if everything is correct, contact IBM Software Support.
- 00208: At least one of the token names was not found in the token list. Check control file setup; if everything is correct, contact IBM Software Support.
- 00212: The control file could not be opened. Check control file setup; if everything is correct, contact IBM Software Support.

User response:

Check control file setup if the description of the error code requires it; if not, or if the control file is correct, contact IBM Software Support.

CKZG2806I **The log apply process will begin at RBA=*rba*'.**

Explanation:

The log apply process will start at the RBA indicated in the message.

User response:

No action is required.

CKZG2812E **A mismatch between passed Zparm information and the JES SSCT was found.**

Explanation:

This is an internal error indicating that the Zparm array that is being passed to Db2 Cloning Tool is inconsistent with the subsystem list found inside MVS data structures.

User response:

Contact IBM Software Support.

CKZG2813I **The log reader process will launch a total of *nnnn* tasks.**

Explanation:

This message indicates the total number of tasks that will be launched.

User response:

No action is required.

CKZG2814I **The log reader will launch total of 1 task per member.**

Explanation:

Indicates that processing of the log reader will launch a total of one task per member since PARALLEL has been set to 0.

User response:

No action is required.

CKZG2815I **The log reader process will start with PARALLEL tasks = *nnnn***

Explanation:

The log reader process will start with the indicated maximum number of tasks.

User response:

No action is required.

CKZG2816I ***task_number* The log reader task #*task_number* finished.**

Explanation:

Indicates that processing of the log reader finished.

User response:

No action is required.

CKZG2817E **The log reader task init failed. RC=*X*"*return_code*"**

Explanation:

The log apply processing failed to initialize a task necessary for reading logs. The reason code is specified in the error message.

User response:

Contact IBM Software Support.

CKZG3000E **The space *database.table_space PART partition* has an unknown space status.**

Explanation:

The status of the space indicated in the message is not known.

User response:

When Db2 Cloning Tool checks the space to see if it is in recover pending, a status code unknown to Db2 Cloning Tool was found. Contact IBM Software Support.

CKZG3001E **The stop status check for space *database.table_space PART partition* timed out.**

Explanation:

The stop status check for the space indicated in the message timed out.

User response:

After the repair operation is started, Db2 Cloning Tool checks the space and waits for the recover pending flag to be removed by Db2. This message was generated because after checking 5 times in 15 seconds, the space was still in recover pending status. You must remove the recover pending status manually.

CKZG3002E **An attempt to Repair the Recover Pending status failed.**

Explanation:

The JCL attempted to repair the recover pending status but the repair failed.

User response:

When Db2 Cloning Tool called Db2 to repair the recover pending status for the space, the operation finished with an error condition. Contact IBM Software Support.

CKZG3003E **An error occurred on an attempt to open the DSNUTILB Steplib.**

Explanation:

Db2 Cloning Tool was unable to open the DSNUTILB Steplib.

User response:

The Db2 loadlib concatenation specified on the **Update DB2 Subsystem Parameters** panel is incomplete. When DSNUTILB attempted to use this concatenation, some of the required load modules were not found. Verify that you have specified the correct load modules on the **Update DB2 Subsystem Parameters** panel. If the problem persists, contact IBM Software Support.

CKZG3004E **The Repair operation's SYSPRINT output dataset could not be opened.**

Explanation:

Db2 Cloning Tool was unable to open the repair operation's SYSPRINT output data set.

User response:

Verify that the data set exists and is available for use.

CKZG3005E **The Repair operation's SYSIN dataset allocation failed.**

Explanation:

Db2 Cloning Tool was unable to allocate the repair operation's SYSIN data set.

User response:

To call Db2 to repair the recover pending status, a SYSIN data set must be allocated to hold the Db2 command stream. Db2 Cloning Tool was unable to allocate a SYSIN data set. Check the settings you specified in the User Settings option and correct any errors.

CKZG3006E **Dynamic allocation return code = *return_code***

Explanation:

This message reports the return code associated with the failed dynamic allocation attempt.

User response:

Ensure the data set exists and is available for use.

CKZG3007E **The SYSIN DD could not be opened for output during Repair processing.**

Explanation:

Db2 Cloning Tool was unable to open the SYSIN DD during repair processing.

User response:

Check the settings you specified in the User Settings option and correct any errors.

CKZG3008E **Open error code=*error_code***

Explanation:

This message reports the open error code that Db2 Cloning Tool encountered when it attempted to open the SYSIN DD.

User response:

Check the settings you specified in the User Settings option and correct any errors.

CKZG3009E **The Repair operation's SYSPRINT dataset allocation failed.**

Explanation:

Db2 Cloning Tool was unable to allocate the repair operation's SYSPRINT data set.

User response:

To call Db2 to repair the recover pending status, a SYSPRINT data set needs to be allocated to hold the Db2 command processor's output stream. Db2 Cloning

Tool was unable to allocate this data set. Check the settings you specified in the User Settings option and correct any errors.

CKZG3260W **Incremental method SORT is obsolete. MERGE mode used instead.**

Explanation:
INCREMENTAL SORT is no longer supported (it is ignored). The internal method used instead is MERGE.

User response:
No action is required.

CKZG3401E **The following XML SSID/DBname/ TSname control card is invalid:**

Explanation:
The control cards do not conform to expected syntax.

User response:
Correct the JCL and resubmit the job.

CKZG3402I *message_text*

Explanation:
This message is generated with CKZG3401E.

User response:
No action is required.

CKZG3405E **Could not obtain SSID and User Indicator from input parameters.**

Explanation:
The log apply job was unsuccessful in trying to connect to the specified subsystem when processing spaces with XML data.

User response:
Verify that the subsystem SSID specified in the job is accurate. Correct the JCL and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3406E **Could not open the SYSOUT DD.**

Explanation:
The log apply job could not open the SYSOUT DD.

User response:
Verify that the SYSOUT DD is specified in the job. Correct the JCL and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3407E **Invalid SYSOUT DD LRECL.**

Explanation:
The LRECL specified on the SYSOUT DD is incorrect.

User response:
Verify that the LRECL specified in the SYSOUT DD is accurate. Correct the JCL and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3450I **Object *object* required no action.**

Explanation:

The object was determined to require no action to make the object usable.

User response:
No action is required.

CKZG3451I **Object *object* had its sequence nbr increased by *rowcount*.**

Explanation:
Db2 Cloning Tool updated the catalog to make the XML object usable.

User response:
No action is required.

CKZG3452I **With a source count=*count***

Explanation:
Db2 Cloning Tool updated the catalog to make the XML object usable.

User response:
No action is required.

CKZG3453E **The XML sequence number could not be obtained for source object.**

Explanation:
Coordination of the internal XML sequence number during OBIDXLAT processing could not be completed.

User response:
IBM Software Support

CKZG3500E **The XML target SSID/DBname/ TSname control cards are invalid.**

Explanation:
The subsystem, database name or table space name are invalid in the log apply control cards.

User response:
Correct the subsystem, database name or table space name and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3501I **The SPACE(...) set involved that the error was detected in was *spacesetnumber***

Explanation:
Indicates the SPACE set involved in the error.

User response:
No action is required.

CKZG3502E **The XML sequence number could not be obtained for source object.**

Explanation:
Coordination of the internal XML sequence number during OBIDXLAT processing could not be completed.

User response:
IBM Software Support

CKZG3604I **The SPACE(...) set involved that the error was detected in was spaceset**

Explanation:

This error message is produced with other error messages and indicates the name of the space that is causing the error.

User response:

Check other messages for processing errors.

CKZG3605E **The end point for database.table_space did not match the UNIFIED value.**

Explanation:

The end point for the table space indicated in the message did not match the value specified for the UNIFIED value.

User response:

No action is required.

CKZG3606E **Consistency value = X'consistency_token'**

Explanation:

A problem occurred with the UNIFIED or UNIFIED_WARNING control cards. The message displays the consistency value taken from the first object in the group.

User response:

The values shown in CKZG3606E and CKZG3607E can be compared for diagnostic purposes. The value shown in CKZG3606E is that for the first object in the group, while the value shown in CKZG3607E is the value for any object that does not match it. For example, if there are 10 objects in the group and three do not match the first, then one CKZG3606E message will display with three CKZG3607E messages (for each group).

CKZG3607E **Object's derived value = X'consistency_token'**

Explanation:

A problem occurred with the UNIFIED or UNIFIED_WARNING control cards. The message displays the object's derived value for the first object in the group. This message displays any object that does not match the first object in the group (identified in CKZG3606E).

User response:

The values shown in CKZG3606E and CKZG3607E can be compared for diagnostic purposes. The value shown in CKZG3606E is that for the first object in the group, while the value shown in CKZG3607E is the value for any object that does not match it. For example, if there are 10 objects in the group and three do not match the first, then one CKZG3606E message

will display with three CKZG3607E messages (for each group).

CKZG3609E **The resource optimization phase [1|2] of DB2 sort failed. RC='return_code'**

Explanation:

Db2 Sort failed in the initialization step necessary for optimization.

User response:

Contact IBM Software Support.

CKZG3611E **There is not enough storage to perform the desired number of parallel sorts.**

Explanation:

There is not enough virtual storage space to perform the number of parallel sorts specified.

User response:

Lower the number of parallel tasks specified or increase the amount of virtual memory specified available for the job, then resubmit the job.

CKZG3612E **Insufficient total storage to perform the desired number of parallel sorts.**

Explanation:

The amount of storage available for a sort was insufficient.

User response:

Increase your region size or reduce the number of parallel tasks, then resubmit the job.

CKZG3711E **Syntax error around TO_TIMESTAMP value. Form is "YYYY-MM-DD-HH.MM.SS.XXXXXX"**

Explanation:

There is a syntax error in the value of TO_TIMESTAMP. The valid format is "YYYY-MM-DD-HH.MM.SS.XXXXXX"

User response:

Correct the JCL and resubmit the job.

CKZG3784E **The GMT OFFSET parameter was specified, but no value was found with it**

Explanation:

The GMT offset value is empty.

User response:

Add the GMT-OFFSET parameter in the format +/-hh:mm, where hh is hours in decimal value and mm is minutes in decimal value.

CKZG3785E **Invalid GMT OFFSET value**

Explanation:

The GMT value specified does not confirm to the specification +/-hh:mm where *hh* is hours in decimal value and *mm* is minutes in decimal value.

User response:

Enter a valid value as described in the message text.

CKZG3786E GMT OFFSET keyword has already been coded

Explanation:

The GMT OFFSET has already been specified in the job.

User response:

Remove the duplicate GMT-OFFSET keyword.

CKZG3787E SKIP-LOG-APPLY keyword has already been coded for this run.

Explanation:

The SKIP-LOG-APPLY value has already been specified for this run.

User response:

Remove the duplicate keyword and re-run the job.

CKZG3788I Db2 Cloning Tool will skip log read and log apply for this run.

Explanation:

The SKIP_LOG_APPLY keyword has been enabled for this run. Db2 Cloning Tool will skip log read and log apply for this run. The target will have the same data consistency as the image copies that are selected as a source.

User response:

No action is required.

CKZG3795W If SKIP_LOG_APPLY is specified, UNIFIED or UNIFIED WARNING keywords are ignored.

Explanation:

The UNIFIED or UNIFIED_WARNING keywords was specified in the batch job, but SKIP_LOG_APPLY is also enabled. When SKIP_LOG_APPLY is enabled, there is no guarantee that all the objects will have a unified end point. Therefore, the UNIFIED or UNIFIED_WARNING is not valid and is ignored.

User response:

No action is required.

CKZG3801E The conversion program returned an error.

Explanation:

This message is the header line for additional messages that follow.

User response:

No action is required.

CKZG3802E The Db2 Cloning Tool row conversion program ended unexpectedly.

Explanation:

An error occurred in the Db2 Cloning Tool row conversion program.

User response:

This message is accompanied by related messages that provide more information about the issue. If you cannot resolve the issue, note the job return code and contact IBM Software Support.

CKZG3806I Begin Log Apply Summary Report

Explanation:

This message marks the beginning of the Log Apply Summary Report.

User response:

No action is required.

CKZG3807I End Log Apply Summary Report

Explanation:

This message marks the end of the Log Apply Summary Report.

User response:

No action is required.

CKZG3830E An error occurred when processing CKZG exit call function: *function*

Explanation:

This message indicates that an error occurred when processing the following function calls: Initialize, Open data set and Close data set.

User response:

Check additional error messages. Contact IBM Software Support if the problem persists.

CKZG3831I Calling page exit function: *function*

Explanation:

This informational message indicates that one of the following function calls are issued: Initialize, Open dataset and Close dataset.

User response:

No action is required.

CKZG4600E Unrecognized function symbol *symbol*

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG4601E Failed to get the number of physical fragments. The MMGR driver call finished with

MMGR_EXTENT_COUNT_NA return value.

Explanation:

An internal error occurred when calculating the number of extents for VSAM data set.

User response:

Contact IBM Software Support.

**CKZG4602E Incorrect input: SpaceType = char
char hex hex**

Explanation:

An invalid value was specified for space type.

User response:

Contact IBM Software Support.

CKZG4603E Incorrect input: DSSIZE = dssize

Explanation:

An invalid data set size was specified.

User response:

Contact IBM Software Support.

**CKZG4604E Incorrect input: PageSize =
pagesize**

Explanation:

An invalid page size was specified.

User response:

Contact IBM Software Support.

**CKZG4605E Incorrect input: SpecifiedPriQTY =
priqty**

Explanation:

An invalid primary allocation size was specified.

User response:

Contact IBM Software Support.

**CKZG4606E Incorrect input:
SpecifiedPriQTYCylinders =
cylinders**

Explanation:

An invalid primary allocation size was specified.

User response:

Contact IBM Software Support.

**CKZG4607E Incorrect input: SpecifiedSecQTY
= secqty**

Explanation:

An invalid value for the secondary allocation size was specified.

User response:

Contact IBM Software Support.

**CKZG4608E Incorrect input:
SpecifiedSecQTYCylinders =
cylinders**

Explanation:

An invalid value was specified for the data set secondary allocation size.

User response:

Contact IBM Software Support.

CKZG4609E Incorrect input: TSQTY = tsqty

Explanation:

An invalid value was specified for the primary allocation quantity for the table space.

User response:

Contact IBM Software Support.

CKZG4610E Incorrect input: IXQTY = ixqty

Explanation:

An invalid value was specified for the primary allocation quantity for the index space.

User response:

Contact IBM Software Support.

**CKZG4611E The dataset *data set* has reached
the maximum size *size* GB. There
is no more space available.**

Explanation:

The data set is too small to hold all of the data.

User response:

Recreate the object with a larger data set size.

**CKZG4612E Failed to open dataset *data set*
Could not allocate buffering area
(RC = rc, SYSRC = sysrc, SYSRSN =
sysrsn).**

Explanation:

An error occurred during the allocation of a pagefixed buffer.

User response:

Contact IBM Software Support.

**CKZG4613E Failed to open dataset *data set*
Could not fix buffering area (RC =
rc, RSN = rsn).**

Explanation:

An error occurred when allocating the pagedfixed buffer.

User response:

Contact IBM Software Support.

**CKZG4614E Failed to open dataset *data set*.
The MMGR driver RC = rc**

Explanation:

An error occurred when opening the data set for an object.

User response:

Contact IBM Software Support.

CKZG4615E **Failed to write data in dataset *data set*. RC = *rc*, RSN = *rsn*, MMGR RC = *rc*, MMGR RSN = *rsn***

Explanation:

An internal error occurred when attempting to write data to the indicated data set.

User response:

Contact IBM Software Support.

CKZG4616E **Failed to close dataset *data set*. RC = *rc***

Explanation:

The indicated data set could not be closed.

User response:

Contact IBM Software Support.

CKZG4617E **The buffering area pointer is NULL on close before deallocation of dataset *data set***

Explanation:

The pointer to the allocated memory was corrupted.

User response:

Contact IBM Software Support.

CKZG4618E **The buffering area not unfix on close before deallocation of dataset *data set***

Explanation:

An internal error occurred.

User response:

Contact IBM Software Support.

CKZG4619E **The buffering area not freed on close before deallocation of dataset *data set* RC = *rc*, SYSRC = *sysrc*, SYSRSN = *sysrsn***

Explanation:

The allocated memory can't be freed.

User response:

Contact IBM Software Support.

CKZG4620E **The connection pointer is NULL before read from dataset.**

Explanation:

Incorrect connection pointer initialization value.

User response:

Contact IBM Software Support.

CKZG4621E **The buffering area pointer is NULL before read from dataset.**

Explanation:

The buffering area was not initialized correctly.

User response:

Contact IBM Software Support.

CKZG4622E **The read/write buffer size is 0 before read from dataset.**

Explanation:

There is no allocated memory.

User response:

Contact IBM Software Support.

CKZG4623E **The buffering area size *size* bytes is not enough to store read/write buffer size *size* bytes.**

Explanation:

The buffering area is insufficient to store all of the data.

User response:

Contact IBM Software Support.

CKZG4624E **The read/write buffer pointer is NULL.**

Explanation:

The pointer to the allocated buffer area was corrupted.

User response:

Contact IBM Software Support.

CKZG4625E **Failed to read data from dataset *data set* RC = *rc*, RSN = *rsn*, MMGR RC = *rc*, MMGR RSN = *rsn***

Explanation:

An error occurred while reading data from the data set.

User response:

Contact IBM Software Support.

CKZG4626W **EOF reached at read from dataset *data_set_name*. Number of bytes read *data_size***

Explanation:

An error occurred during the read process. The end of data set was reached while reading data.

User response:

Contact IBM Software Support.

CKZG4627E **The connection pointer is NULL before write to dataset.**

Explanation:

The connection pointer initialization value is invalid.

User response:

Contact IBM Software Support.

CKZG4628E **The buffering area pointer is NULL before write to dataset.**

Explanation:

The buffering area was not initialized correctly.

User response:

Contact IBM Software Support.

CKZG4629E The read/write buffer size is 0 before write to dataset.

Explanation:

There is insufficient allocated memory.

User response:

Contact IBM Software Support.

CKZG4630E Failed to write data to dataset data set RC = rc, RSN = rsn, MMGR RC = rc, MMGR RSN = rsn

Explanation:

An error occurred while writing data to the data set.

User response:

Contact IBM Software Support.

CKZINP01E ERROR PARSING PARMLIB INI# MEMBER ON LINE n | ERROR PARSING PARMLIB INI MEMBER ON LINE n

Explanation:

An error was found in the CKZINI member. The line shows the current line being processed.

User response:

Look for other messages and repair the CKZINI member.

CKZINP02E UNMATCHED */ | Continued statement at end of file

Explanation:

An end-comment (*/) was found for which there was no begin-comment (/*).

User response:

Remove the end-comment or insert a begin-comment in the appropriate location.

CKZINP05E TOKEN NAME LONGER THAN 72

Explanation:

A token name longer than 72 bytes was found on the specified line.

User response:

Reduce the length of the token to 72 or less.

CKZINP06E EXPECTED = AFTER TOKEN NAME

Explanation:

An equal sign (=) was not found after the token name.

User response:

Insert an equal sign (=) between the token and its value.

CKZINP07E EXPECTED = AFTER TOKEN NAME

Explanation:

An equal sign (=) was not found after the token name.

User response:

Insert an equal sign (=) between the token and its value.

CKZINP08E FOUND NON-NOTES TOKEN BEFORE 1ST SECTION NAME

Explanation:

A token was found before any Section was specified. Only the "Notes[®]" token is allowed before a section name. Reminder: Notes is provided for the customer and no product code can access the Notes value.

User response:

Ensure that the first non-comment line and non-Notes token is a Section name.

CKZINP09E DUPLICATE SECTION/TOKEN FOUND | section | token

Explanation:

A token cannot be defined twice for the same section.

User response:

Remove the redundant token and retry.

CKZINP10E file function FAIL, RC = nnnnnnnn

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZINP11E Open failure, DD ddname | DD INI OPEN FAILURE | DD SYSPRINT open failure | DD UPDATE open failure | DD TRACEDD open failure

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZINP12E EXPECTED PARM VALUE OR RECORD, FOUND function

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZINP14E ERROR PARSING MSCINI

Explanation:

MSCINI INIMERGE failed to parse the MSCINI input because it could not find the INIMERGE_SECTION values.

User response:

Use the INI# member distributed by IBM and retry. If you need assistance, contact IBM Software Support.

CKZINP15W COLUMNS 73-80 NOT BLANK ON LINE

Explanation:

INI parser detected characters in columns 73-80. IBM has found that many INI errors occur because the person editing the INI does not see text in columns 73-80.

User response:

Edit the PARMLIB INI member so that columns 73-80 are blank.

CKZINP21E SYSPLEX NAME MORE THAN 8 CHARACTERS

Explanation:

The sysplex value on an INI section statement is invalid because it is too long.

User response:

Repair the section name and retry.

CKZINP22E SYSTEM NAME MORE THAN 8 CHARACTERS

Explanation:

The system value on an INI section statement is invalid because it is too long.

User response:

Repair the section name and retry.

CKZINP31E SECTION NAME LONGER THAT 72

Explanation:

The section name is limited in size.

User response:

Repair the section name and retry.

CKZINP32E INVALID SECTION NAME

Explanation:

The section name can have alphanumeric characters in addition to a period, dash, underscore or slash.

User response:

Repair the section name and retry.

CKZINP33E PERIOD(".") FOUND IN INI# SECTION

Explanation:

A period character in the section name is valid in a customer INI for denoting sysplex and system names, but is not valid in the distributed INI#.

User response:

Repair the section name and retry.

CKZINP34E QUALIFIED INIMERGE_VALUES IN INI | QUALIFIED PRODUCT_INFO IN INI

Explanation:

These sections cannot be qualified by sysplex and/or system name.

User response:

Repair the section name and retry.

CKZINP35E TWO INIMERGE_VALUES SECTIONS FOUND

Explanation:

The INIMERGE_VALUES may only appear once in the INI.

User response:

Repair the section name and retry.

CKZINP36E QUALIFIED SECTION AFTER GENERIC

Explanation:

A section was found with sysplex and/or system qualifications but it follows the same section without such qualification. This section can never be accessed and is invalid.

User response:

Repair the section name and retry.

CKZINP37E DUPLICATE QUALIFIED SECTION | DUPLICATE UNQUALIFIED SECTION FOUND

Explanation:

Two section names match exactly, then are either 1) both unqualified or 2) both specify the same Sysplex and system names. Section names must be unique.

User response:

Repair the section name and retry.

CKZINP38E TRAILING PERIOD ON SECTION STATEMENT

Explanation:

A section statement has one of the following formats: :sectionname.sysplexname.systemname :sectionname.sysplexname :sectionname..systemname

User response:

Repair the section name and retry.

CKZINP39E INI# COLUMNS 73-80 ARE NOT BLANK, LINE nnnn | INI COLUMNS 73-80 ARE NOT BLANK, LINE nnnn

Explanation:

Historically, placing text in columns 73-80 makes debugging errors in the INI difficult.

User response:

Repair the section name and retry.

CKZINP40E RIGHT OF INI#"S SECTION NOT BLANK

Explanation:

INI#'s section statement must be all blanks after section name to facilitate customers entering sysplex/system names.

User response:

Repair the section name and retry.

CKZINP41I NOTHING COPIED**Explanation:**

The customer INI had no special sections to copy.

User response:

No action is required.

CKZINP42I NOTHING DELETED**Explanation:**

No INI parameters were deleted.

User response:

No action is required.

CKZINP50I INPUT: input line**Explanation:**

A previously noted error occurred on the line shown.

User response:

See prior error message.

CKZINP51E TOKEN NON-NOTES BEFORE 1ST SECTION**Explanation:**

NOTES is the only token allowed before the first section statement.

User response:

Repair the INI and resubmit.

CKZINP52E TOKEN NOTES FOUND IN INI#**Explanation:**

NOTES is only allowed in the customer INI.

User response:

Repair the INI and resubmit.

**CKZINP53E ERROR PARSING MSCINI (INI#),
SEE JOB LOG | ERROR PARSING
THE CUSTOMER INI, SEE JOB LOG****Explanation:**

INIMERGE used the INI parser to validate the INI but the INI parser was unsuccessful.

User response:

For the customer INI, repair the INI based on the messages in the JOB log and resubmit. For INI#, contact IBM Software Support.

CKZINP71E SECTION IN INI# BUT NOT INI:**Explanation:**

INIVIEW cannot find section in INI that was found in the INI#.

User response:

Repair the INI and resubmit.

CKZINP72I ALL INI# SECTIONS FOUND IN INI**Explanation:**

All of the INI# sections were found in the INI.

User response:

No action is required.

CKZINP73E ERROR IN SYSPARM**Explanation:**

CKZ01INV SYSPARM is incorrect. Valid forms are: SYSPARM= for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM='plexname,sysname'

User response:

Correct the SYSPARM on your execute statement and resubmit.

**CKZINS01E CKZ01VV1 END FAILURE, RC =
nnnnnnnn****Explanation:**

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINS02E PASSED SECTION LENGTH
INVALID length****Explanation:**

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINS03E PASSED TOKEN LENGTH INVALID
length****Explanation:**

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZINS04E INIGET 3RD PARM, NO VL BIT**Explanation:**

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINS05E INI TOKEN REC SHORTER THAN
KEY SECTION/TOKEN section/
token****Explanation:**

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINS06E TOKEN VALUE LONGER THAN
RECEIVING PARM SECTION/
TOKEN token TOKEN VALUE value****Explanation:**

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZINS07E BAD REG 2 PTR TO VECTOR TABLE

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINV00I CKZ01INV STARTING (REV=nn,
PMR=xxxxxxx,
FIXDATE=xxxxxxx)**

Explanation:

Message shows current version of the module.

User response:

No action is required.

**CKZINV01E ERROR PARSING PARMLIB INI#
MEMBER ON LINE n | ERROR
PARSING PARMLIB INI MEMBER
ON LINE n**

Explanation:

An error was found in when parsing an INI member.

User response:

Look for subsequent messages and repair the PARMLIB(xxxINI) or PARMLIB(xxxINI#) member as indicated.

**CKZINV02E UNMATCHED */ | CONTINUED
STATEMENT AT END OF FILE**

Explanation:

An end-comment (*/) was found for which there was no begin-comment (/*).

User response:

Remove the end-comment or insert a begin-comment in the appropriate location.

CKZINV05E TOKEN NAME LONGER THAN 72

Explanation:

A Token name longer than 72 bytes was found on the specified line.

User response:

Reduce the length of the token to 72 or less.

CKZINV06E EXPECTED = AFTER TOKEN NAME

Explanation:

An equal sign (=) was not found after the Token name.

User response:

Insert an equal sign (=) between the Token and its value.

CKZINV07E EXPECTED = AFTER TOKEN NAME

Explanation:

An equal sign (=) was not found after the Token name.

User response:

Insert an equal sign (=) between the Token and its value.

**CKZINV08E FOUND NON-NOTES TOKEN
BEFORE 1ST SECTION NAME**

Explanation:

A Token was found before any Section was specified. Only the "Notes" Token is allowed before a section name. Reminder: Notes is provided for the customer and no product code can access the notes value.

User response:

Ensure that the first non-comment line and non-Notes token is a Section name.

**CKZINV09E DUPLICATE SECTION/TOKEN
FOUND | Section | Token**

Explanation:

A token can not be defined twice for the same section.

User response:

Remove the redundant token and retry.

CKZINV10E file function FAIL, RC = nnnnnnnn

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINV11E OPEN FAILURE, DD ddname | DD
INI OPEN FAILURE | DD SYSPRINT
OPEN FAILURE | DD UPDATE OPEN
FAILURE | DD TRACEDD OPEN
FAILURE**

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZINV12E EXPECTED PARM VALUE OR
RECORD, FOUND function**

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZINV14E ERROR PARSING MSCINI

Explanation:

MSCINI INIMERGE failed to parse the MSCINI input because it could not find the INIMERGE_SECTION values.

User response:

Use the INI# member distributed by IBM Software Support and retry. If you need assistance, contact IBM Software Support.

CKZINV15W COLUMNS 73-80 NOT BLANK ON LINE

Explanation:

INI parser detected characters in columns 73-80. IBM Software Support has found that many INI errors occur because the person editing the INI doesn't see text in columns 73-80.

User response:

Edit the PARMLIB INI member so that columns 73-80 are blank.

CKZINV21E SYSPLEX NAME MORE THAN 8 CHARACTERS

Explanation:

The sysplex value on an INI section statement is invalid because it is too long.

User response:

Repair the section name and retry.

CKZINV22E SYSTEM NAME MORE THAN 8 CHARACTERS

Explanation:

The system value on an INI section statement is invalid because it is too long.

User response:

Repair the section name and retry.

CKZINV31E SECTION NAME LONGER THAT 72

Explanation:

The section name is limited in size.

User response:

Repair the section name and retry.

CKZINV32E INVALID SECTION NAME

Explanation:

The section name can have alphanumeric characters in addition to a period, dash, underscore or slash.

User response:

Repair the section name and retry.

CKZINV33E Period(".") found in INI# section

Explanation:

A period character in the section name is valid in a customer INI for denoting sysplex and system names but is not valid in the distributed INI#.

User response:

Repair the section name and retry.

CKZINV34E QUALIFIED INIMERGE_VALUES IN INI | QUALIFIED PRODUCT_INFO IN INI

Explanation:

These sections can not qualified by sysplex and/or system name.

User response:

Repair the section name and retry.

CKZINV35E TWO INIMERGE_VALUES SECTIONS FOUND

Explanation:

The INIMERGE_VALUES may only appear once in the INI.

User response:

Repair the section name and retry.

CKZINV36E QUALIFIED SECTION AFTER GENERIC

Explanation:

A section was found with sysplex and/or system qualifications but it follows the same section without such qualification. This section can never be accessed and is invalid.

User response:

Repair the section name and retry.

CKZINV37E DUPLICATE QUALIFIED SECTION | DUPLICATE UNQUALIFIED SECTION FOUND

Explanation:

Two section names match exactly, then are either 1) both unqualified or 2) both specify the same sysplex and system names. Section names must be unique.

User response:

Repair the section name and retry.

CKZINV38E TRAILING PERIOD ON SECTION STATEMENT

Explanation:

A section statement has one of the following formats: :sectionname.sysplexname.systemname :sectionname.sysplexname :sectionname..systemname

User response:

Repair the section name and retry.

CKZINV39E INI# COLUMNS 73-80 ARE NOT BLANK, LINE nnnn | INI COLUMNS 73-80 ARE NOT BLANK, LINE nnnn

Explanation:

Historically, placing text in columns 73-80 makes debugging errors in the INI difficult.

User response:

Repair the section name and retry.

CKZINV40E RIGHT OF INI#'S SECTION NOT BLANK

Explanation:

INI#'s section statement must be all blanks after section name to facilitate customers entering sysplex/system names.

User response:

Repair the section name and retry.

CKZINV41I NOTHING COPIED

Explanation:

The customer INI had no special sections to copy

User response:

No action is required.

CKZINV42I NOTHING DELETED

Explanation:

No INI parameters were deleted.

User response:

No action is required.

CKZINV50I INPUT: input line

Explanation:

A previously noted error occurred on the line shown.

User response:

See prior error message.

CKZINV51E TOKEN NON-NOTES BEFORE 1ST SECTION

Explanation:

NOTES is the only token allowed before the first section statement.

User response:

Repair the INI and resubmit.

CKZINV52E TOKEN NOTES FOUND IN INI#

Explanation:

NOTES is only allowed in the customer INI.

User response:

Repair the INI and resubmit.

**CKZINV53E ERROR PARSING MSCINI (INI#),
SEE JOB LOG | ERROR PARSING
THE CUSTOMER INI, SEE JOB LOG**

Explanation:

INIMERGE used the INI parser to validate the INI but the INI parser was unsuccessful.

User response:

For INI#, contact IBM Software Support. For the customer INI, repair the INI based on the messages in the JOB log and resubmit.

CKZINV71E SECTION IN INI# BUT NOT INI:

Explanation:

INIVIEW cannot find section in INI that was found in the INI#.

User response:

Repair the INI and resubmit.

CKZINV72I ALL INI# SECTIONS FOUND IN INI

Explanation:

All of the INI# sections were found in the INI.

User response:

No action is required.

CKZINV73E ERROR IN SYSPARM

Explanation:

CKZ01INV SYSPARM is incorrect. Valid forms are: SYSPARM= for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM='plexname,sysname'

User response:

Correct the SYSPARM on your execute statement and resubmit.

**CKZMER01E ERROR PARSING PARMLIB INI#
MEMBER ON LINE n | ERROR
PARSING PARMLIB INI MEMBER
ON LINE n**

Explanation:

An error was found in the CKZINI member. The line shows the current line being processed.

User response:

Look for other messages and repair the CKZINI member.

**CKZMER02E UNMATCHED */ | Continued
statement at end of file**

Explanation:

An end-comment (*/) was found for which there was no begin-comment (/).

User response:

Remove the end-comment or insert a begin-comment in the appropriate location.

CKZMER05E TOKEN NAME LONGER THAN 72

Explanation:

A token name longer than 72 bytes was found on the specified line.

User response:

Reduce the length of the token to 72 or less.

CKZMER06E EXPECTED = AFTER TOKEN NAME

Explanation:

An equal sign (=) was not found after the token name.

User response:

Insert an equal sign (=) between the token and its value.

CKZMER07E EXPECTED = AFTER TOKEN NAME

Explanation:

An equal sign (=) was not found after the token name.

User response:

Insert an equal sign (=) between the token and its value.

**CKZMER08E FOUND NON-NOTES TOKEN
BEFORE 1ST SECTION NAME**

Explanation:

A token was found before any Section was specified. Only the "Notes" token is allowed before a section name. Reminder: Notes is provided for the customer and no product code can access the Notes value.

User response:

Ensure that the first non-comment line and non-Notes token is a Section name.

**CKZMER09E DUPLICATE SECTION/TOKEN
FOUND | section | token**

Explanation:

A token cannot be defined twice for the same section.

User response:

Remove the redundant token and retry.

CKZMER10E file function FAIL, RC = nnnnnnnn

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZMER11E DD INI OPEN FAILURE | DD
SYSPRINT open failure | DD
UPDATE open failure | DD
TRACEDD open failure**

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

**CKZMER12E EXPECTED PARM VALUE OR
RECORD, FOUND function**

Explanation:

An unrecoverable error occurred during processing.

User response:

Contact IBM Software Support.

CKZMER14E ERROR PARSING MSCINI

Explanation:

MSCINI INIMERGE failed to parse the MSCINI input because it could not find the INIMERGE_SECTION values.

User response:

Use the INI# member distributed by IBM and retry. If you need assistance, contact IBM Software Support.

**CKZMER15W COLUMNS 73-80 NOT BLANK ON
LINE**

Explanation:

INI parser detected characters in columns 73-80. IBM has found that many INI errors occur because the person editing the INI does not see text in columns 73-80.

User response:

Edit the PARMLIB INI member so that columns 73-80 are blank.

**CKZMER21E SYSPLEX NAME MORE THAN 8
CHARACTERS**

Explanation:

The sysplex value on an INI section statement is invalid because it is too long.

User response:

Repair the section name and retry.

**CKZMER22E SYSTEM NAME MORE THAN 8
CHARACTERS**

Explanation:

The system value on an INI section statement is invalid because it is too long.

User response:

Repair the section name and retry.

CKZMER31E SECTION NAME LONGER THAT 72

Explanation:

The section name is limited in size.

User response:

Repair the section name and retry.

CKZMER32E INVALID SECTION NAME

Explanation:

The section name can have alphanumeric characters in addition to a period, dash, underscore or slash.

User response:

Repair the section name and retry.

**CKZMER33E PERIOD(".") FOUND IN INI#
SECTION**

Explanation:

A period character in the section name is valid in a customer INI for denoting sysplex and system names, but is not valid in the distributed INI#.

User response:

Repair the section name and retry.

**CKZMER34E QUALIFIED INIMERGE_VALUES IN
INI | QUALIFIED PRODUCT_INFO
IN INI**

Explanation:

These sections cannot be qualified by sysplex and/or system name.

User response:

Repair the section name and retry.

CKZMER35E TWO INIMERGE_VALUES SECTIONS FOUND

Explanation:

The INIMERGE_VALUES may only appear once in the INI.

User response:

Repair the section name and retry.

CKZMER36E QUALIFIED SECTION AFTER GENERIC

Explanation:

A section was found with sysplex and/or system qualifications but it follows the same section without such qualification. This section can never be accessed and is invalid.

User response:

Repair the section name and retry.

CKZMER37E DUPLICATE QUALIFIED SECTION | DUPLICATE UNQUALIFIED SECTION FOUND

Explanation:

Two section names match exactly, then are either 1) both unqualified or 2) both specify the same Sysplex and system names. Section names must be unique.

User response:

Repair the section name and retry.

CKZMER38E TRAILING PERIOD ON SECTION STATEMENT

Explanation:

A section statement has one of the following formats: :sectionname.sysplexname.systemname :sectionname.sysplexname :sectionname..systemname

User response:

Repair the section name and retry.

CKZMER39E INI# COLUMNS 73-80 ARE NOT BLANK, LINE nnnn

Explanation:

Historically, placing text in columns 73-80 makes debugging errors in the INI difficult.

User response:

Repair the section name and retry.

CKZMER40E RIGHT OF INI#'S SECTION NOT BLANK

Explanation:

INI#'s section statement must be all blanks after section name to facilitate customers entering sysplex/system names.

User response:

Repair the section name and retry.

CKZMER41I NOTHING COPIED

Explanation:

The customer INI had no special sections to copy.

User response:

No action is required.

CKZMER42I NOTHING DELETED

Explanation:

No INI parameters were deleted.

User response:

No action is required.

CKZMER50I INPUT: input line

Explanation:

A previously noted error occurred on the line shown.

User response:

See prior error message.

CKZMER51E TOKEN NON-NOTES BEFORE 1ST SECTION

Explanation:

NOTES is the only token allowed before the first section statement.

User response:

Repair the INI and resubmit.

CKZMER52E TOKEN NOTES FOUND IN INI#

Explanation:

NOTES is only allowed in the customer INI.

User response:

Repair the INI and resubmit.

CKZMER53E ERROR PARSING MSCINI (INI#), SEE JOB LOG | ERROR PARSING THE CUSTOMER INI, SEE JOB LOG

Explanation:

INIMERGE used the INI parser to validate the INI but the INI parser was unsuccessful.

User response:

For INI#, contact IBM Software Support. For the customer INI, repair the INI based on the messages in the JOB log and resubmit.

CKZMER71E SECTION IN INI# BUT NOT INI:

Explanation:

INIVIEW cannot find section in INI that was found in the INI#.

User response:

Repair the INI and resubmit.

CKZMER72I ALL INI# SECTIONS FOUND IN INI

Explanation:

All of the INI# sections were found in the INI.

User response:

No action is required.

CKZMIF00E Cannot open input *input_ddname*

Explanation:

An input DD cannot be opened.

User response:

If unable to determine the reason the DD open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZMIF01E No data set name of load modules in first parameter in input *input_ddname*

Explanation:

A data set name is invalid or not specified in the input data set defined by *input_ddname*.

User response:

Enter a valid data set name in the input data set.

CKZMIF02E No load module names in input *input_ddname*

Explanation:

No load module names were entered in the input data set that is defined by the *input_ddname*.

User response:

Enter at least one load module name in the input data set. Each load module name should be in a separate input string.

CKZMIF03E Insufficient RAM to allocate *buffer_size* bytes

Explanation:

The required buffer size cannot be allocated.

User response:

If unable to determine the reason for this error, contact IBM Software Support. Have available the listing that contains this message.

CKZMIF04E Cannot open output *output_ddname*

Explanation:

An output DD cannot be opened.

User response:

If unable to determine the reason the DD open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZMIF05E Load module cannot be opened

Explanation:

A load module cannot be opened.

User response:

If unable to determine the reason the load module open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZMIF06E Insufficient RAM to reallocate internal buffer to *buffer_size* bytes

Explanation:

The required buffer size cannot be reallocated.

User response:

If unable to determine the reason for this error, contact IBM Software Support. Have available the listing that contains this message.

CKZMIF07E Incorrect object module information at load module address *load_module_address*

Explanation:

An object module contains incorrect compiling information.

User response:

Contact IBM Software Support.

CKZMIF08E Incorrect return code (RC) of *utility_name* utility: *reason_code*. Please contact IBM Software Support

Explanation:

A utility returned an incorrect return code.

User response:

Contact IBM Software Support.

CKZMIF09W Warnings occurred. For more information, see the content of output *output_ddname*

Explanation:

Warnings occurred during processing of load modules. Reasons for the warnings are described in the output data set defined by *output_DDname*.

User response:

To determine the appropriate response, refer to the content of the output data set defined by the output DD name.

CKZMIF10E Errors occurred. For more information, see the content of output *output_ddname*

Explanation:

Errors occurred during processing of load modules. Reasons for the errors are described in the output data set defined by *output_DDname*.

User response:

To determine the appropriate response, refer to the content of the output data set defined by the output DD name.

CKZMOD01I **Starting**
 (rev=xxx,ptf=xxx,fixdate=xxx)

Explanation:
 Indicates CKZ01MOD is starting and its version.

User response:
 Informational only

CKZMOD02I **Terminating**

Explanation:
 Indicates CKZ01MOD is terminating.

User response:
 Informational only

CKZMOD02E **DIFF option requires SYSUT2**

Explanation:
 DIFF compares loadlibs SYSUT1 and SYSUT2.

User response:
 Correct JCL or execution parameter

CKZMOD03E **Unable to open SYSUT1**

Explanation:
 SYSUT1 is a required DD statement.

User response:
 Correct JCL or execution parameter

CKZMOD04E **A duplicate PMR found module**

Explanation:
 The same PMR number was used twice in the same module.

User response:
 Informational only

CKZMOD05E **No PMR data available for 'SYSUT1
 xxxxxxxx' | No PMR data available
 for 'SYSUT2 xxxxxxxx'**

Explanation:
 The load module has an EHDR, not MSCHDR/MSCPMR macros.

User response:
 Informational only

CKZMOD06E **No Header data available for
 'SYSUT1 xxxxxxxx' | No Header
 data available for 'SYSUT2
 xxxxxxxx'**

Explanation:
 The load module does not have MSCHDR/MSCPMR or EHDR macros.

User response:
 Informational only

CKZMOD30E **FATAL ERROR IN CKZ01VV1
 TABLE(table) FUNC(func) RC(rc)**

Explanation:
 An error occurred using an CKZ01VV1 table.

User response:
 Please report this message to IBM Software Support.

CKZMOD91E **Error READ_PDSE ERROR -
 terminating**

Explanation:
 An error occurred reading a PDSE (not a PDS)

User response:
 Please report this message to IBM Software Support.

CKZMOD92E **Error READ_PDS ERROR -
 terminating**

Explanation:
 An error occurred reading a PDS (not a PDSE)

User response:
 Please report this message to IBM Software Support.

CKZPG000E **Dynamic allocation error. Request:
 request, RC: return_code, RS:
 reason_code**

Explanation:
 An error occurred during dynamic allocation.

User response:
 Check the job log and the system log for messages that might indicate the reason for the failure. If unable to resolve the problem, contact IBM Software Support.

CKZPG001E **VSAM Error. Macro: 'OPEN', ACB
 open flags: open_flags ACB error
 flags: error_flags, RPL feedback:
 feedback_word**

Explanation:
 An error occurred while opening a VSAM data set.

User response:
 Check the job log and the system log for messages that might indicate the reason for the failure. If unable to resolve the problem, contact IBM Software Support.

CKZPG002E **VSAM Error. Macro: 'macro', R15:
 return_code, Feedback:
 feedback_word**

Explanation:
 An error occurred while processing a VSAM data set.

User response:
 Check the job log and the system log for messages that might indicate the reason for the failure. If unable to resolve the problem, contact IBM Software Support.

CKZPG003E **VSAM Error. Macro: 'CLOSE', R15:
 return_code**

Explanation:

An error occurred while closing a VSAM data set.

User response:

Check the job log and the system log for messages that might indicate the reason for the failure. If unable to resolve the problem, contact IBM Software Support.

CKZPG004W Unexpected *field_name* on page *page_number*. Received value: *value_from_data_set*, expected value: *expected_value*

Explanation:

When page *page_number* was processed, the field *field_name* had the value *value_from_data_set*. Translation rules indicate that it should have the value *expected_value*. Processing continues.

User response

If the failure occurred in target job, verify that:

- The current target job run is not a rerun. To rerun the target job, you must rerun all previous cloning jobs, or specify the runtime repository.
- If the source job was run with DATA-MOVER PGM(NONE) and the data set copy was performed outside of Db2 Cloning Tool, verify that the data set copy successfully completed.

If unable to resolve the problem, contact IBM Software Support.

CKZPG005W No translation rule for *field_name* value *value_from_data_set* on page *page_number*

Explanation:

When page *page_number* was processed, the field *field_name* had the value *value_from_data_set* but no translation rule exists for this value. Processing continues.

User response

If the failure occurred in target job, verify that:

- The current target job run is not a rerun. To rerun the target job, you must rerun all previous cloning jobs, or specify the runtime repository.
- If the source job was run with DATA-MOVER PGM(NONE) and the data set copy was performed outside of Db2 Cloning Tool, verify that the data set copy successfully completed.

If unable to resolve the problem, contact IBM Software Support.

CKZPG006W Unable to recognize the type of page *page_number*. PGFLAGS=*flags*

Explanation:

When page *page_number* was processed, its type was not recognized. Processing continues.

User response

If the failure occurred in target job, verify that:

- The data set copy completed without errors.
- Db2 Cloning Tool is compatible with your version of Db2.

If unable to resolve the problem, contact IBM Software Support.

CKZPG007E Media Manager error. Request=*request* RC=*return_code*, RS=*reason_code*

Explanation:

A Media Manager call ended with an error.

User response:

Check the job log and the system log for messages that might indicate the reason for the failure. If unable to resolve the problem, contact IBM Software Support.

CKZPG008I Fast processing will be used for data set *data_set_name*

Explanation:

This informational message indicates that the processing of *data_set_name* does not require updating every page.

User response:

No action is required.

CKZPG009I Regular processing will be used for data set *data_set_name*

Explanation:

This informational message indicates that the processing of *data_set_name* cannot be optimized.

User response:

No action is required.

CKZPG010E Data set is empty

Explanation:

This message indicates that the VSAM data set is empty.

User response:

Verify that the data set copy completed without errors. If unable to resolve the problem, contact IBM Software Support.

CKZPG012E Extent requested and SECQTY is 0

Explanation:

There is no free space in the data set and the target Db2 object was created with SECQTY=0, which disallows data set extensions.

User response:

Re-create the target Db2 objects using other PRIQTY or SECQTY values, and resubmit the cloning jobs. If unable to resolve the problem, contact IBM Software Support.

CKZPG014E **Attempt to apply logs past max allowed page.**
RP=requested_pager_number,
MAP=maximum_available_page_number

Explanation:

This message indicates an internal error.

User response:

Contact IBM Software Support.

CKZPP100I **text**

Explanation:

Message produced when command input is read.

User response:

No action is required.

CKZPP101E **Input file not open**

Explanation:

Message produced when command input is parsed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZPP102E **Input file LRECL invalid**

Explanation:

Message produced when command input is parsed. This error typically occurs if SYSIN DD statement refers to a data set with LRECL other than 80.

User response:

Correct the command input and resubmit.

CKZPP103E **Blank record invalid here**

Explanation:

Message produced when command input is parsed.

User response:

Correct the command input and resubmit.

CKZPP104E **Expected continuation not found**

Explanation:

Message produced when command input is parsed.

User response:

Correct the command input and resubmit.

CKZPP105W **Input flushed**

Explanation:

Message produced when command input is parsed and a prior error has been detected.

User response:

Review the prior CKZPP1 error messages. Correct the command input and resubmit.

CKZPP106E **Unmatched parenthesis**

Explanation:

Message produced when command input is parsed.

User response:

Correct the command input and resubmit.

CKZPP107E **Paren nesting error**

Explanation:

Message produced when command input is parsed.

User response:

Correct the command input and resubmit.

CKZPP108E **Line buffer full**

Explanation:

Message produced when command input is parsed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZPP109E **String delimiters not balanced**

Explanation:

Message produced when command input is parsed. Second quotation character not found.

User response:

Correct the command input and resubmit.

CKZPP110E **Input record buffer length invalid**

Explanation:

Message produced when command input is parsed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE20I **Command ended**

Explanation:

Message produced when command input is parsed and a prior error has been detected.

User response:

Review the prior CKZPSE error messages. Correct the command input and resubmit.

CKZPSE21E **Keyword not found : text**

Explanation:

Message produced when command input is parsed. The displayed keyword is not valid for this command or parent keyword.

User response:

Review the prior CKZPSE error messages. Correct the command input and resubmit.

CKZPSE22E **Module not found: module-name**

Explanation:

Message produced when command input is parsed.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE23E Operand not supported for: text**Explanation:**

Message produced when command input is parsed. The displayed keyword is not implemented.

User response:

Correct the command input and resubmit.

CKZPSE24E Operand truncated for:**Explanation:**

Message produced when command input is parsed. The value specified for the displayed keyword is longer than the keyword accepts.

User response:

Correct the command input and resubmit.

CKZPSE25E Empty or missing parm list**Explanation:**

Message produced when command input is parsed and a prior error has been detected.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE26W Command abandoned**Explanation:**

Message produced when command input is parsed and a prior error has been detected.

User response:

Review the prior CKZPSE error messages. Correct the command input and resubmit.

CKZPSE28E Operand required for keyword: keyword**Explanation:**

Message produced when command input is parsed. The displayed keyword requires a value.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE29E Conflicting keywords specified with keyword:**Explanation:**

Message produced when command input is parsed. Two mutually exclusive keywords have been specified.

User response:

Correct the command input and resubmit.

CKZPSE30E Multiple use of keyword not allowed**Explanation:**

Message produced when command input is parsed. Two mutually exclusive keywords have been specified.

User response:

Correct the command input and resubmit.

CKZR X101E CKZRNTGT ddname READ FAILED, RC IS nn**Explanation:**

The Rexx program was unable to read a required file.

User response:

Add the appropriate DD statement to the execution JCL.

CKZR X102E CKZRNTGT COUNTS DO NOT MATCH: CKZIN = nnn; NUCIN = nnn**Explanation:**

There is an unequal number of records in the file from Db2 Cloning Tool and the file from Db2 Cloning Tool NOUSERCATALOGS.

User response:

Verify that the input files correspond to the correct runs for the two COPYS.

CKZR X103I CKZRNTGT ddname HEADER IS header information**Explanation:**

Informational message.

User response:

No action is required.

CKZR X104I CKZRNTGT ddname INPUT FILE HAS nnn VOLUME PAIRS**Explanation:**

Informational message.

User response:

No action is required.

CKZR X105E CKZRNTGT NEWTGT OPEN FAILED, RC IS nn**Explanation:**

The REXX program was unable to open a required file.

User response:

Add the appropriate DD statement to the execution JCL.

CKZR X106E CKZRNTGT NEWTGT WRITE FAILED, RC IS nn**Explanation:**

The REXX program was unable to add a record to the file.

User response:

Check that the DCB attributes for the output file are correct. Check that there is sufficient space given to the output file.

CKZR107E CKZRNTGT NO MATCH FOUND FOR CKZIN ENTRY source target

Explanation:

No match was found for the indicated Db2 Cloning Tool source and target volume serials in the Db2 Cloning Tool NOUSERCATALOGS input file.

User response:

Verify that the input files correspond to the correct runs for the two COPYs.

CKZR108E CKZRNTGT VERSION MISMATCH: CKZIN: version NUCIN: version

Explanation:

The version entries in the two files do not match.

User response:

Verify that the input files correspond to the correct runs for the two COPYs.

CKZR109E CKZRNTGT VERSION IS INCORRECT, version

Explanation:

The version entry in the input files is not supported.

User response:

Verify that the input files correspond to the correct runs for the two COPYs.

CKZR110E CKZRNTGT CKZIN PRODUCT IS INCORRECT, product

Explanation:

The product entry in the Db2 Cloning Tool input file is incorrect.

User response:

Verify that the Db2 Cloning Tool input file has not been modified.

CKZR111E CKZRNTGT NUCIN PRODUCT IS INCORRECT, product

Explanation:

The product entry in the Db2 Cloning Tool NOUSERCATALOGS input file is incorrect.

User response:

Verify that the Db2 Cloning Tool NOUSERCATALOGS input file has not been modified.

CKZR112E CKZRNTGT NO INPUT READ

Explanation:

No records were found in the input files.

User response:

Verify that the input files correspond to the correct runs for the two COPYs and that the DD statements have not been dummied.

CKZR306E modid Write to CMDOUT failed; return code = return_code

Explanation:

The REXX program was unable to write to the output file allocated to the CMDOUT DD statement. Processing terminates.

User response:

Ensure that the DCB attributes for the CMDOUT file are correct. Ensure that there is sufficient space that is given to the output file. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZR321I modid Using DB2 SSID= ssid

Explanation:

The REXX program will use the indicated DB2 ssid. The ssid value is obtained from the argument string that is provided to the REXX program.

User response:

No action is required.

CKZR322E modid DSN command failed; return code= return_code

Explanation:

The invocation of the DB2 DSN command processor failed. *return_code* is the return code from the invocation.

User response:

Verify that the correct DB2 ssid is being used, that the job is running on the same system as the DB2 system, and that the correct STEPLIB is being used. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZR323I modid Number of commands written to CMDOUT: number_of_commands

Explanation:

The number of start commands that were written to CMDOUT DD statement.

User response:

No action is required.

CKZR324I modid There are no objects with UT status

Explanation:

There are no DB2 objects found that have UT status, therefore no start commands need to be generated.

User response:

No action is required.

CKZR325I *modid* Start commands generated for:**Explanation:**

This message precedes a list of the *database.spacename* objects that have UT status, and that start commands have been generated for.

User response:

No action is required.

CKZR399E *modid* Exec error condition has occurred: *error_data***Explanation:**

The REXX program had an internal error. *error_data* identifies the statement in error. Processing terminates.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

CKZTCZ01E *modid* Discover exec error condition has occurred: *error_information***Explanation:**

An error occurred during the execution of the Discover EXEC. Information about the error is also displayed.

User response:

If unable to determine the reason for the failure from the *error_information*, contact IBM Software Support. Have available a copy of the message and the error information.

CKZTCZ02E *modid* Discover EXEC invoked with no parameters.**Explanation:**

The Discover EXEC was invoked with no parameters.

User response:

Ensure the Discover EXEC was executed from the Tools Customizer Discover Customized Product Information panel. If unable to determine the reason for the failure, contact IBM Software Support.

CKZTCZ03E *modid* Discover EXEC found empty CKZINI member.**Explanation:**

The Discover EXEC found no records in the CKZINI member.

User response:

Ensure the correct data set name was specified and that the data set contains a valid CKZINI member.

CKZTCZ04E *modid* Data set *data_set_name* has no CKZINI member.**Explanation:**

The indicated data set does not have a CKZINI member.

User response:

Ensure the correct data set name was specified and that the data set contains a valid CKZINI member.

CKZTCZ05E *modid* Data set *data_set_name* not usable for Discover: *reason***Explanation:**

The indicated data set is not usable because of the indicated reason.

User response:

Ensure the correct data set name was specified.

CKZVE201E NO INIT FUNCTION RECEIVED.**Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

CKZVE202E INIT FUNCTION RECEIVED WITHOUT INTERVENING TERM**Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

CKZVE203E UNKNOWN FUNCTION: xxxx**Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

CKZVE204E ERROR CALLING CKZ01VV1 xxxx r15=rc r0=rea id=nn**Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

CKZVE205E MLA VALUE FOUND GREATER THAN 4**Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

CKZVE206E MASK FAILED EVALUATION CHECK: xxxx**Explanation:**

Mask xxxx is invalid.

User response:

Fix the error and retry the function.

**CKZVE207E CSI FAILED WHEN CALLING FOR
MASTER CATALOG NAME****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE208E MASK FAILED EVALUATION
CHECK: xxxx - yyyy****Explanation:**

Mask xxxx is invalid for reason yyyy

User response:

Fix the error and retry the function.

**CKZVE209E UN-ALLOCATION FAILED FOR:
cccc****Explanation:**

Un -allocation failed for catalog cccc.

User response:

Determine the reason for and fix the problem, Retry the function.

CKZVE210E ALLOCATION FAILED FOR: cccc**Explanation:**

Allocation failed for catalog cccc.

User response:

Determine the reason for the allocation failure and retry.

**CKZVE211E UCBLook ERROR FOR: VVVV -
RETURN CODE:RC REASON
CODE:RSN****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE212E VOLUME NOT MOUNTED: vvvv
DSN=nnn DEVTYPE:ttt****Explanation:**

A catalog entry specified that a data set was cataloged to a volume, vvvv, that was not mounted for dsn nnn on device type ttt.

User response:

Mount the volume, uncatalog the data set, or respecify the selection masks. Retry the function.

**CKZVE213E IOSCAPU ERROR FOR: VVVV -
RETURN CODE:RC REASON
CODE:RSN****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE214E nnnn : VOLSER=vvvv
RC=rrr,REAS=sss,DESC=zzzz****Explanation:**

An error occurred when trying to access the VVDS on volume vvvv for data set nnnn . rrr = return code. sss = reason code. zzzz = description when available.

User response:

If unable to determine the cause of the error or to fix it, contact IBM Software Support for assistance.

**CKZVE215E UCBCPIN ERROR FOR: VVVV -
RETURN CODE:RC REASON
CODE:RSN****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE216E CVAF ERROR: RETURN CODE:rc
STATUS CODE:sc VOLUME:vvvv
ID=nn PREV FUNC:ffff****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE217E ERROR OBTAINING VTOC DATA
FROM vvvv : nnnn****Explanation:**

An error occurred when trying to access the VTOC on volume vvvv for data set nnnn.

User response:

If unable to determine the cause of the error or to fix it, contact IBM Software Support for assistance.

CKZVE218E HSM MCVT NOT FOUND**Explanation:**

An error occurred when trying to access the HSM control blocks. The most probable cause is HSM has not been started.

User response:

If unable to determine the cause of the error or to fix it, contact IBM Software Support for assistance.

**CKZVE219E HSM DATA EXTRACT ENDED WITH
ERRORS****Explanation:**

An error occurred when trying to access the HSM control blocks. The most probable cause is HSM has not been started.

User response:

If unable to determine the cause of the error or to fix it, contact IBM Software Support for assistance.

**CKZVE220E MIGRATED DATASET NOT FOUND
 IN MCDS: nnnn****Explanation:**

The catalog indicated that data set nnnn was migrated, but no entry was found in the HSM MCDS.

User response:

If unable to determine the cause of the error or to fix it, contact IBM Software Support for assistance.

**CKZVE221W NO CATALOG ENTRIES FOUND FOR
 MASK(S)****Explanation:**

No catalog entries were found for any of the masks passed to CKZ01VE2.

User response:

If this result was not expected, try to determine the reason for the problem and retry the function. If help is needed, contact IBM Software Support for assistance.

**CKZVE222E ASSOCIATED ENTRY NOT FOUND
 IN CATALOG - nnnn****Explanation:**

A non-VSAM alias entry with an association of nnnn. nnnn was not found in the catalog. The most probable cause is an orphaned catalog entry.

User response:

Determine the reason for and fix the problem. Retry the function. If help is needed, contact IBM Software Support for assistance.

**CKZVE223E UNEXPECTED CATALOG RECORD
 CONSTRUCT - ASSOCIATION CELL
 NOT WHERE EXPECTED****Explanation:**

Catalog entry for entry mentioned is invalid.

User response:

Determine the reason for and fix the problem. Retry the function. If help is needed, contact IBM Software Support for assistance.

**CKZVE224E TYPE "J" RECORD WITH NO
 MATCHING "B" RECORD: nnnn****Explanation:**

Catalog entry nnnn is an extension record for a catalog record that does not exist.

User response:

Determine the reason for and fix the problem. Retry the function. If help is needed, contact IBM Software Support for assistance.

**CKZVE225E ERROR READING SELF
 DESCRIBING RECORD FROM
 CATALOG: nnnn****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE226E RECEIVED A "GET" REQUEST
 BEFORE RECEIVING A MASK****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE227E INPUT CONTROL BLOCK IS NOT
 COMPATIBLE****Explanation:**

This is an internal error.

User response:

Contact IBM Software Support.

**CKZVE229E HSM IS NOT ACTIVE - UNABLE TO
 RETRIEVE 'REQUESTED DATA****Explanation:**

HSM must be active for MCDS data to be retrieved.

User response:

Start HSM and retry the function.

**CKZVE230I ALLOCATION FAILED FOR: nnnn -
 WILL NOT BE USED FOR SEARCH****Explanation:**

Unable to allocate catalog nnnn. No entries that may reside in this catalog will be retrieved.

User response:

Informational only. Processing continues.

**CKZVE231E VOLUME NOT MOUNTED: vvvv
 DSN=nnn DEVTYPE:ttt****Explanation:**

A catalog entry specified that a data set was cataloged to a volume, vvvv, that was not mounted for dsn nnn on device type ttt.

User response:

Mount the volume, uncatalog the data set, or respecify the selection masks. Retry the function.

**CKZVE232E NO MCDS DATASET WITH VALID
 KEY-RANGE FOUND****Explanation:**

An attempt was made to find a dsn in the HSM MCDS but no MCDS found in the HSM address space had a key-range that would accommodate the dsn.

User response:

Processing continues. No MCDS data will be included for this data set.

**CKZVE233E NO MCDS DSN(S) FOUND IN HSM
 ADDRESS SPACE**

Explanation:

An attempt was made to find the data set name(s) of the MCDS(s) allocated to HSM. The attempt failed.

User response:

Processing continues. No MCDS data will be included for this data set.

**CKZVE234W ERROR CALLING CKZ01HSM
RC=RC**

Explanation:

This is an internal error.

User response:

Processing continues. No MCDS data will be included. If the MEMBER NAME MSCKZVE2 reason for the failure cannot be determined, contact IBM Software Support.

**CKZVE235E nnnn : VOLSER=vvvvvv - VVDS
NOT OPEN**

Explanation:

A previous open failed for the VVDS on volume vvvvvv failed.

User response:

Processing continues. No VVDS related data is returned for data set nnnn.

**CKZVE236W n: Invalid DSN found :
XXXXXXXXXXXXXXXXXXXX**

Explanation:

The catalog contained an invalid data set name. The first CKZVE236W shows the data set name in character format. The next three CKZVE236W show the name in hex.

User response:

Fix the catalog and retry the function.

**CKZVE237W OPEN FAILED FOR: nnnn - WILL
NOT BE USED FOR SEARCH**

Explanation:

Unable to open catalog nnnn. No entries that may reside in this catalog will be retrieved.

User response:

Informational only. Processing continues.

**CKZVE238E CLUSTER NOT FOUND FOR
TRUENAME IN CATALOG -**

Explanation:

An error has been detected in your catalog. A VSAM true name record was found without an associated cluster entry.

User response:

Informational only. Processing continues. Evaluate catalog entries for the identified data set.

**CKZVE239I RETRYING ERROR
OBTAINING ??????? FOR dsn -
ucat**

Explanation:

An error occurred when trying to access the VTOC, VVDS, or MCB information for a data set. This message will occur the catalog is incorrect. This message may occur if the data set is being migrated so that the catalog is temporarily out of date.

User response:

No action required for this message unless subsequent errors CKZVE217E, CKZCA120E, CKZCA114E, CKZCA220E, or CKZCA214E See action for these messages.

**CKZVE251E type error; VOLSER: volser CCHHR:
cccchhhrr CCHH_LO: cccchhhh
CCHH_HI: cccchhhh POST: xx Loc:
llll**

Explanation:

An error occurred when trying to access or process the VVDS on the indicated volume. For type of EXCP, an error was returned from a read of a VVDS block. Other types indicate an expected record was not found in the VVDS. lllll is an internal indicator of where the problem occurred.

User response:

Contact IBM Software Support. Have available the listing that contains this message.

**CKZVSE00E VSAM FAILURE,DDNAME: ddname
VSAM DSN: datasetname or NO DD
ALLOCATION FOUND**

Explanation:

A VSAM failure occurred attempting to open, access, or close a VSAM data set used by Db2 Cloning Tool.

User response:

See additional messages. Although many errors could be caused by internal errors, look for any messages with ** USER CORRECTABLE **. These could include problems such as insufficient region size, unable to allocate extents, etc. Some errors could be the result of incorrectly defining a data set. Compare values returned for catalog LRECL, key length, and key offset to the specifications for the data set as documented in the appropriate product manual. LRECL and KEYLEN to be compared are those supplied with messages CKZVSE21 and CKZVSE22, not CKZVSE19 and CKZVSE20. If the error does not appear to be user correctable, please include all CKZVSEnn messages in documentation supplied to IBM Software Support. Also supply to IBM Software Support the output from a LISTCAT ENT(...) ALL for the failing data set.

CKZVSE01E **FAILING CKZ01VSI FUNCTION:
nnn and PREVIOUS CKZ01VSI
FUNCTION: nnn**

Explanation:

Last and previous (if any) VSAM function performed.

User response:

Message provided for product debugging. However, other messages could indicate a user correctable situation. Note that this is the last logical request made by a product program. Because of implicit opens and closes, see message CKZVSE13E to determine the exact VSAM function last requested when determining which set of IBM documented return/reason codes apply.

CKZVSE03E **PROGRAM CSECT: csect name**

Explanation:

Csect name of failing program.

User response:

Message provided for debugging. However, other messages could indicate a user correctable situation.

CKZVSE04E **VWHEN ERROR ID: id**

Explanation:

Internally assigned id of last VSAM I/O call.

User response:

Message provided for debugging. However, other messages could indicate a user correctable situation.

CKZVSE05E **VWHEN ASM LISTING LINE #: line
number**

Explanation:

Source listing line number of last VSAM I/O call.

User response:

Message provided for debugging. However, other messages could indicate a user correctable situation.

CKZVSE06E **message associated with
CKZ01VSI Register 15 value**

Explanation:

message associated with a non-VSAM error - i.e. a non-zero return code from CKZ01VSI or a VSAM OPEN, ACCESS, or CLOSE.

User response:

Message provided for debugging. However, other messages could indicate a user correctable situation. If last function is OPEN, and return code is 16: RLS NOT AVAILABLE - NO SMSVSAM SERVER this may be user correctable if the CKZINI member specified RLS for the data set and in fact RLS is not supported on the image.

CKZVSE07E **message associated with VSAM
Register 15 value**

Explanation:

message associated with VSAM Register 15 value.

User response:

See message CKZVSE10 and CKZVSE11 for VSAM reason code. and description. If CKZVSE12 is produced instead of CKZVSE11, see the IBM manual for a description of the reason code.

CKZVSE08E **SVC99 ERROR CODE: code**

Explanation:

Error code from SVC99 - dynamic allocation.

User response:

Use this error code in conjunction with the SVC99 information code (CKZVSE09E) to determine the cause of failure.

CKZVSE09E **SVC99 INFORMATION CODE: code**

Explanation:

Information code from SVC99 - dynamic allocation.

User response:

Use this error code in conjunction with the SVC99 error code (CKZVSE08E) to determine the cause of failure.

CKZVSE10E **VSAM REASON CODE: code**

Explanation:

Reason code returned from VSAM open, access, or close.

User response:

Use this error code in conjunction with the description provided by message CKZVSE11E to determine the cause of the failure. See additional comments for CKZVSE11E.

CKZVSE11E **vsam reason code description**

Explanation:

Abbreviated description for selected VSAM reason codes.

User response:

The error handling module producing CKZVSEnn messages includes many VSAM reason code descriptions. Note that these messages are abbreviated in comparison to the messages as documented by IBM. For this reason, users should also refer to the IBM documentation for the reason code displayed with message CKZVSE10. Because IBM documented reason codes are distinguished by open/close versus access, see message CKZVSE13E for the last access attempted. CKZVSE13E will indicate OP for open, and CL for close. Consider any other value as ACCESS. Note especially any reason code descriptions with the string ** USER CORRECTABLE as these may be situations correctable without IBM Software Support.

CKZVSE12E **NO DESCRIPTION FOR REASON
CODE**

Explanation:

Description for reason code not included in error handling table.

User response:

Refer to the description of the reason code (CKZVSE10E) for the last function requested (CKZVSE13E) in IBM documentation.

CKZVSE13E VSAM CCODE: code

Explanation:

This is a value indicating the last VSAM request issued.

User response:

This value is necessary if looking up the return and reason codes in IBM documentation. CCODE will be OP for an open and CL for a close. Consider any other value as ACCESS.

CKZVSE14E CKZ01VSI IMPLICIT OPEN

Explanation:

Indicator that last program request required that the data set be re-opened.

User response:

Primarily for product debugging.

CKZVSE15E CKZ01VSI IMPLICIT CLOSE

Explanation:

Indicator that a close was the result of one task losing control to another.

User response:

Primarily for product debugging.

CKZVSE16E CKZ01VSI IMPLICIT REPOSITION

Explanation:

Indicator that the positioning had to be re-established as a result of losing control to another task.

User response:

Primarily for product debugging.

CKZVSE17E OPEN CLASSIFICATION: value

Explanation:

Internal value indicating intended use of the data set.

User response:

Primarily for product debugging.

CKZVSE18E BUFFERING TECHNIQUE: value

Explanation:

Internal value indicating selected buffering technique.

User response:

Primarily for product debugging.

CKZVSE19E CKZ01VSI PARM LRECL: lrecl

Explanation:

Last LRECL set in CKZ01VSI parm field.

User response:

Primarily for product debugging.

CKZVSE20E CKZ01VSI PARM KEYLEN: key length

Explanation:

Last key length set in CKZ01VSI parm field.

User response:

Primarily for product debugging.

CKZVSE21E CKZ01VSI CATALOG LRECL: lrecl

Explanation:

Max LRECL fetched from catalog at open time.

User response:

This should be used to verify that the cluster as defined matches the requirements for the data set as defined in product documentation.

CKZVSE22E CKZ01VSI CATALOG KEYLEN: key length

Explanation:

Key length fetched from catalog at open time.

User response:

This should be used to verify that the cluster as defined matches the requirements for the data set as defined in product documentation.

CKZVSE23E CKZ01VSI CATALOG KEY OFFSET: key offset

Explanation:

Key offset fetched from catalog at open time.

User response:

This should be used to verify that the cluster as defined matches the requirements for the data set as defined in product documentation.

CKZVV136E Invalid LRECL: value

Explanation:

LRECL must be a positive integer.

User response:

correct the LRECL value.

CKZVV137E Dataspace defined over 2 GIG

Explanation:

Dataspaces are limited to 2 GIG or 2097152 K. The corresponding SI027_VALUES in CKZINI are specified in K, so the limit is 2097152.

User response:

Correct the SI027_Values section of the CKZINI

CKZVV138E Error creating dataspace rc=xx reason yyyyyyyy

Explanation:

CKZ01VV1 attempted to create a dataspace. The error indicates that the parameters were incorrect.

User response:

Correct the SI027_Values section of the CKZINI

CKZVV139E **Error extending dataspace rc=xx
reason yyyyyyyy**

Explanation:

CKZ01VV1 attempted to expand a dataspace. The error indicates that the parameters were incorrect.

User response:

Correct the SI027_Values section of the CKZINI

CKZVV140E **Bad Token: INITIAL_SIZE**

Explanation:

Value specified is larger than the value allowed (2097152).

User response:

Correct the INITIAL_SIZE value in the CKZINI

CKZVV141E **Bad Token: MAXIMUM_SIZE**

Explanation:

Value specified is larger than the value allowed (2097152).

User response:

Correct the MAXIMUM_SIZE value in the CKZINI

CKZVV142E **Bad Token:
MAXIMUM_DATASPACE_SIZE**

Explanation:

Value specified is larger than the value allowed (2097152).

User response:

Correct the MAXIMUM_DATASPACE_SIZE value in the CKZINI.

How to look up message explanations

You can use several methods to search for messages and codes.

Searching an information center

In the search box that is located in the top left toolbar of any Eclipse help system, such as the IBM Information Management Software for z/OS Solutions Information Center, enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use the asterisk wildcard character (*) to represent multiple characters, and you can use the question mark wildcard character (?) to represent a single character.

The information center contains the latest message information for all of the Information Management products that are included in the information center.

Searching for messages on the web

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you are presented with links to the message information in IBM information centers.

Gathering diagnostic information

Before you report a problem with Db2 Cloning Tool to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Db2 Cloning Tool problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of Db2 that you are using and the type and version of the operating system that you are using

- For Db2 Cloning Tool Table Space Cloning, if copied objects are inaccessible on the target, provide the create DDL and DSN1PRNT of the page or pages in error.

Provide additional information based on the type of problem that you experienced:

For errors in batch processing, provide the following information:

- The complete source, target, source TCP/IP server, and target TCP/IP server job logs for Db2 Cloning Tool Table Space Cloning, including SYSUDUMP DD output.



Attention: The SYSUDUMP DD is used only when an abend occurs. At a minimum, it is recommended to specify the following parameters for SYSUDUMP in PARMLIB:

```
OPTIONS SDATA=(SUM),
PDATA=(SA,REGS,LPA,JPA,PSW,SPLS,SUBTASKS)
```

All of these parameters should be specified to ensure that when an abend occurs there is enough data for the first data capture to facilitate diagnosis.

To control the size of the job output, use the JOB card, the JES2 /*JOBPARM statement, or the JES3 // *MAIN statement parameters to limit job output.

- The complete job logs for Db2 Cloning Tool Subsystem Cloning.
- DSNJU004 output for each member of the data sharing group for Db2 Cloning Tool Table Space Cloning with enabled log apply.
- SYSLOG output for the time that the failed job was run.
- DSN1PRNT output of the target data set on which processing failed, and DSN1PRNT output of its corresponding source data set (taken as soon as possible after the job failed) for Db2 Cloning Tool Table Space Cloning. The target data set should not have been rewritten by a different test; this is important for accurate analysis. DSN1PRNT output is required only when an error occurred during data set processing.
- The minilog data set that contains records for the failed object (or object's partition) for Db2 Cloning Tool Table Space Cloning with enabled log apply. To determine the required minilog data set, please contact IBM Software Support. Do not rerun the test and do not delete minilog data sets until IBM Software Support obtains the required minilog data set for analysis.
- Contents of any data sets that were used during the processing.

For online abends, provide the following information:

- A screen shot of the panel that you were using when the abend occurred
- The job log from the TSO session that encountered the abend
- A description of the task that you were doing before the abend occurred

Gathering maintenance level information with the DISPLAY MEPL command

You can use the DISPLAY MEPL ISPF interface command to produce a report about maintenance level information for all product load modules. The report contains module name, service level, date, and time information that might be useful when diagnosing software problems.

Procedure

1. On any Db2 Cloning Tool ISPF interface panel, enter DISPLAY MEPL and press Enter.
2. On the **Build Job for Display MEPL Utility** panel, fill out the following fields:
 - **Build Job In:** Specify the data set and member name into which the job will be generated. If the data set does not exist, Db2 Cloning Tool will allocate the data set. You can optionally specify the data set allocations for the new data set by selecting the **Specify new data set allocation parameters** field.

- **Output Sequential Data Set for Display MEPL Utility:** Enter a sequential data set name where the output from the DISPLAY MEPL job will be placed after the job is submitted. If the data set does not exist, it will be created.
 - **Job Cards:** Enter valid job cards for your location.
3. Press Enter.
- If you selected the **Specify new data set allocation parameters** field, the **Data set allocation parameters** window is displayed. Press Enter when you have finished setting the data set allocation parameters.
4. Submit the job that is built in the specified member.
- When the job completes, the output data set contains the maintenance level report.
- A portion of the report is shown below:

| OBJECT | SERVICE | DATE | TIME |
|-----------------------|---------|-------------|----------|
| ----- | | | |
| Load module CKZ00990: | | | |
| CKZF001 | HCKZ320 | Nov 30 2014 | 21:03:09 |
| CKZF002 | HCKZ320 | Nov 30 2014 | 21:03:07 |
| CKZF003 | HCKZ320 | Nov 30 2014 | 21:03:11 |
| CKZF004 | HCKZ320 | Nov 30 2014 | 21:03:13 |
| CKZF005 | HCKZ320 | Nov 30 2014 | 21:03:14 |

Validating load module contents with MODLEVEL

IBM Software Support may request that you submit the MODLEVEL JCL to identify the current modification levels of your product load modules.

MODLEVEL displays internal identifiers that can be used by IBM Software Support to determine the level of the product you are running. This information is based on information stored internally within load modules. Please note that only the last update (level) will contain the fix and revision number and the change date. All other fixes will have only the ID number.

If you experience difficulty running the report, or have any questions, contact IBM Software Support.

JCL for MODLEVEL can be found in member CKZMLVL in the SCKZJCL library. The following is an example of the MODLEVEL JCL.

```
//STEP01 EXEC PGM=CKZ01MOD
//STEPLIB DD DISP=SHR,DSN=CKZ.LOAD (your LOAD library)
//SYSUT1 DD DISP=SHR,DSN=*.STEPLIB
//SYSPRINT DD SYSOUT=*
```

DD statements for the MODLEVEL JCL are as follows:

STEPLIB

Required. Specify the product LOAD library.

SYSPRINT

Required. Output DD statement.

SYSUT1

Required. Specify the primary LOAD library to be processed.

Execution of the MODLEVEL JCL creates a report containing the last fix and a list of fixes for all of the modules in the SYSUT1 library. The following is an example of the Db2 Cloning Tool Subsystem Cloning MODLEVEL report.

```

CKZMOD01I 09/28/2011 13.38.16 DVLP CKZ01MOD Starting (REV=10,ptf=PRM01060,fixdate=2009/07/24)
CKZMOD06E 09/28/2011 13.38.16 DVLP CKZ01MOD No Header data available for 'SYSUT1 CKZ$ACES'
CKZMOD06E 09/28/2011 13.38.16 DVLP CKZ01MOD No Header data available for 'SYSUT1 CKZ$BMSG'
CKZMOD06E 09/28/2011 13.38.16 DVLP CKZ01MOD No Header data available for 'SYSUT1 CKZ$BOOT'

```

Highest Build Date is 2011/08/25 13:52

Highest PMR Date is 2011/04/08 16:40 PMR00629
Highest ETR Date is 2011/08/24 13:34 ETR23082

| Member | DD | Last fix | Vers | Rev | Build Date/Time | Macro |
|-----------|---------------------|---|--------|-----|------------------|-------------|
| CKZ\$ACES | SYSUT1 | ** Non-standard header | | | | |
| CKZ\$BMSG | SYSUT1 | ** Non-standard header | | | | |
| CKZ\$BOOT | SYSUT1 | ** Non-standard header | | | | |
| CKZ00010 | SYSUT1 | LAST ETR00680 | 3 | 32 | 2011/03/09 17:39 | MSCHDR |
| | SYSUT1 | LIST 16362,16057,16397,16514,16739,16290,17907,17456, | | | | |
| | SYSUT1 | LIST 18311,16289,19527,21924,21276,19773,25006,26790, | | | | |
| | SYSUT1 | LIST 26909,26936,00277,00602,00680 | | | | |
| CKZ00011 | SYSUT1 | LAST ETR00868 | V3R1M0 | 16 | 2011/03/09 17:40 | MSCHDR |
| | SYSUT1 | LIST 14608,15131,15194,15738,15948,16323,16973,17280, | | | | |
| | SYSUT1 | LIST 17446,17683,19413,21570,19773,23330,26934,00251, | | | | |
| | SYSUT1 | LIST 00868 | | | | |
| CKZ00500 | SYSUT1 | LAST N/A | @DF | 0 | 2011/08/24 13:32 | MSCHDR |
| CKZ00501 | SYSUT1 | LAST N/A | @02 | 0 | 2011/08/24 13:32 | MSCHDR |
| CKZ01VV1 | SYSUT1 | LAST ETR26666 | 1.1 | 27 | 2011/03/09 10:46 | MSCHDR |
| | SYSUT1 | LIST 09781,10091,10145,11098,11401,11778,12763,13631, | | | | |
| | SYSUT1 | LIST 13631,13873,14582,14584,14585,14740,14864,14877, | | | | |
| | SYSUT1 | LIST 14970,16675,16781,20351,26666 | | | | |
| CKZ01VV2 | SYSUT1 | LAST ETR26636 | 0101 | 6 | 2011/03/09 10:46 | MSCHDR |
| | SYSUT1 | LIST 16384,18817,12226,18917,20082,23548,26636 | | | | |
| CKZMOD02I | 09/28/2011 13.38.20 | DVLP CKZ01MOD | | | | Terminating |

Tools Customizer troubleshooting

Use this information to diagnose and correct problems that you experience with Tools Customizer.

Gathering diagnostic information

Before you report a problem with Tools Customizer to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Tools Customizer problems:

- A clear description of the problem and the steps that are required to re-create the problem
- Relevant screen captures
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of Db2 that you are using and the type and version of the operating system that you are using
- The Tools Customizer trace data set
- The Tools Customizer data store data set and the *high_level_qualifier*.SCCQTENU data set

Determining the trace data set name

You will need to identify the name of the trace data set if you cannot allocate the trace data set, the trace data set runs out of space, or IBM Software Support asks for it.

The name of the trace data set depends on the prefix setting in the TSO profile. To identify the name of the trace data set, you must know the prefix setting.

- If PREFIX is set, the name of the trace data set is *prefix*.CCQ.TRACE, where *prefix* is the TSO prefix that you specified in the profile.
- If NOPREFIX is set, the name of the trace data set is *user_ID*.CCQ.TRACE, where *user_ID* is your TSO user ID.

Chapter 29. Tools Customizer reference

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology and data sets

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Products and components

How an IBM Tool is packaged determines whether it is referred to as a product or as a component in the Tools Customizer documentation and interface. An IBM Tool that is ordered as a stand-alone entity (that is, not as part of a solution pack) is referred to as a product. An IBM Tool that is part of a solution pack is referred to as a component. Some IBM Tools are available in both formats; therefore, the same IBM Tool can be referred to as a product or as a component depending on how it is packaged.

Db2 entry

You can customize Db2 Cloning Tool on one or more Db2 entries. A Db2 entry can be any of the following items:

Db2 subsystem

A distinct instance of a relational database management system (RDBMS) that is not part of a data sharing group. An example of a Db2 subsystem name is DB01.

Db2 group attach name

The name that is used by the TSO/batch attachment, the call attachment facility (CAF), DL/I batch, utilities, and the Resource Recovery Services attachment facility (RRSAF) as a generic attachment name. An example of a group attach name is DSG1.

Db2 data sharing member

A Db2 subsystem that is assigned by the cross-system coupling facility (XCF) to a data sharing group. An example of a Db2 data sharing member name is DB02.

Tools Customizer maintains the following lists of Db2 entries:

Associated list

The list of Db2 entries that are associated with Db2 Cloning Tool. If the product to be customized requires Db2 entries, you can customize Db2 Cloning Tool only on Db2 entries that are in the associated list. When you customize Db2 Cloning Tool, this list is displayed in the DB2 Entries, Associations, and Parameter Status section of the **Customizer Workplace** panel.

You can add and copy Db2 entries to the associated list. When you add or copy Db2 entries to the associated list, the entries are associated with Db2 Cloning Tool.

Master list

The list of all Db2 entries that are defined but are not associated with Db2 Cloning Tool. Tools Customizer obtains information about these Db2 entries either from entries that were created manually or from the customizations of other products that were discovered. If you remove a Db2 entry from the associated list, the Db2 entry is added to the master list. When you create a new Db2 entry, it is added to the master list, and when you associate the new entry with Db2 Cloning Tool, it is removed from the master list and added to the associated list. The master list is displayed on the **Associate a DB2 Entry for Product** panel.

If the associated list does not have the Db2 entries on which you want to customize Db2 Cloning Tool, you can associate existing entries from the master list to the associated list.

You can create new Db2 entries and copy existing entries to the master list.

High-level qualifier

The high-level qualifier is considered to be all of the qualifiers except the lowest level qualifier. A high-level qualifier includes a mid-level qualifier.

Product parameters

Parameters that are specific to Db2 Cloning Tool. These parameters are defined by Db2 Cloning Tool and are stored in a data member that is defined by Db2 Cloning Tool.

LPAR parameters

Parameters on the local LPAR that are required to customize Db2 Cloning Tool. These parameters are defined by Tools Customizer and are stored in an LPAR parameter data member.

Db2 parameters

Parameters for a Db2 entry. These parameters are defined by Tools Customizer and are stored in a Db2 parameter data member.

Status type

Product, LPAR, and Db2 entry status type

After you specify the product that you want to customize, the product, the LPAR, and the Db2 entries have a status. The status is partly based on whether required parameters are defined. For some products, LPAR parameters or Db2 parameters might not be required. In these cases, the status is Not Required.

To customize Db2 Cloning Tool, all of the required parameters must be defined.

If required parameters for the the product parameters, LPAR parameters, or Db2 parameters are not defined, the status of the parameters is Incomplete. Define values for parameters by manually editing them or by generating the customization jobs and specifying values for all of the required parameters that are displayed on the panels.

When values for all of the required parameters are defined, the status is Ready to Customize. Customization jobs can be generated only when all of the required parameters are defined and the status is Ready to Customize or Customized for the product parameters, LPAR parameters, and Db2 parameters for the Db2 entries on which Db2 Cloning Tool will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

Table 93. Status types for the product, the LPAR, and the Db2 entries

| Status | Product | LPAR | Db2 entries |
|--------------------|--|---|---|
| Incomplete | The required product parameters are not defined. | The required parameters are not defined. | The required parameters are not defined. |
| Discovered | The product parameter definitions were discovered by using the product Discover EXEC. | N/A | N/A |
| Ready to Customize | The required product, is Ready to Customize or Customized for the LPAR and at least one associated Db2 entry. You can generate the customization jobs. | The required LPAR parameters are defined or LPAR parameters are not required. | The required Db2 parameters are defined or Db2 parameters are not required. |

Table 93. Status types for the product, the LPAR, and the Db2 entries (continued)

| Status | Product | LPAR | Db2 entries |
|-------------------------|--|---|---|
| Verify Values | The required product or component parameter values are defined but they either have not been verified or verification is not enabled on the Product or Component Parameters panel. | The required LPAR parameter values are defined, but they either have not been verified or verification is not enabled on the LPAR Parameters panel. | The required Db2 parameter values are defined, but they either have not been verified or verification is not enabled on the Db2 Parameters panel. |
| Customized | The jobs are customized on the local LPAR. | The jobs are customized for the product or for all of the associated Db2 entries on the local LPAR. | The jobs are customized for the Db2 entry. |
| Errors in Customization | N/A | N/A | Errors occurred while the customization jobs were being generated. |
| Not Required | N/A | LPAR parameters are not required. | Db2 parameters are not required. |

Related tasks

[Creating and associating Db2 entries](#)

You can create new Db2 entries and associate them with Db2 Cloning Tool.

[Copying Db2 entries](#)

You can copy associated and not associated Db2 entries to other Db2 entries or to new Db2 entries.

[Removing Db2 entries](#)

You can remove Db2 entries from the associated list.

Data sets that Tools Customizer uses during customization

Tools Customizer uses several unique data sets during the customization process. Familiarize yourself with these data sets before you begin to use Tools Customizer.

Several different data sets are required to customize Db2 Cloning Tool with Tools Customizer. These data sets are supplied by Db2 Cloning Tool, supplied by Tools Customizer, or allocated by Tools Customizer.

Db2 Cloning Tool provides the following data sets:

Metadata library

Contains the metadata for the product to be customized. Tools Customizer uses the metadata to determine which tasks, steps, and parameters to display on the **Product Parameters** panel, the **LPAR Parameters** panel, and the **DB2 Parameters** panel. This data set also contains the templates that Tools Customizer uses to generate the customization jobs.

The metadata library naming convention is *high_level_qualifier.SCKZDENU*, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.

You specify the metadata library on the **Specify the Metadata Library** panel. READ access to this data set is required.

Discover EXEC library

Contains the Db2 Cloning Tool Discover EXEC. When you customize Db2 Cloning Tool, you can use the Discover EXEC to automatically retrieve and store product information, such as parameter values from an already customized product. Tools Customizer saves the discovered information in the data store.

The default name of the data set is the high-level qualifier for the metadata library plus a lowest-level qualifier. For Db2 Cloning Tool, the lowest-level qualifier is SCKZDENU. You can change the default value on the **Discover Customized Product Information** panel. EXECUTE access to this data set is required.

Tools Customizer provides the following data sets:

Tools Customizer metadata library

Contains the metadata for the Db2 and LPAR parameters that are required to customize Db2 Cloning Tool. Tools Customizer uses the metadata to determine which parameters to display on the **DB2 Parameters** panel and the **LPAR Parameters** panel. In addition, Tools Customizer uses information in the metadata library to determine whether additional Db2 and LPAR parameters need to be displayed on these panels. As you customize different products, different Db2 and LPAR parameters might need to be defined.

The default name of the data set is DB2TOOL.CCQ110.SCCQDENU. You can change the default value on the **Tools Customizer Settings** panel. READ access to this data set is required.

Tools Customizer table library

Stores information about jobs that are customized. Job information that is stored includes a description of the job, its member name and template name, the SSID, group attach name, and when the job was generated.

The default name of the data set is DB2TOOL.CCQ110.SCCQTENU. WRITE access to this data set is required.

Tools Customizer requires that the following data sets exist during the customization process. If the data sets do not exist, Tools Customizer automatically allocates them.

Discover output data set

Contains the output that is generated when you run the Db2 Cloning Tool Discover EXEC. The Db2 Cloning Tool Discover EXEC retrieves the metadata and values for the parameters from a previous customization of Db2 Cloning Tool.

The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the **Tools Customizer Settings** panel or the **Discover Customized Product Information** panel. WRITE access to this data set is required.

Data store data set

Contains product, LPAR, and Db2 parameter values, and Db2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, Db2 subsystems or data sharing groups, and LPAR when you customize products on the local LPAR.

The default name of the data set is DB2TOOL.CCQ110.DATASTOR. You can change the default value on the **Tools Customizer Settings** panel. WRITE access to this data set is required.

Customization library

Contains the customization jobs that Tools Customizer generates for Db2 Cloning Tool.

Tools Customizer checks whether a customization library name was specified for more than one instance of the same version of the same product. If the same customization library name is specified for more than one product of the same version, the CCQD123E message is issued to prevent you from overwriting previously generated customization jobs. Ensure that you specify unique qualifier for the customization library for each instance of the product.

To customize Db2 Cloning Tool, submit the members of the data set in the order in which they are displayed on the **Finish Product Customization** panel.

The data set naming convention is *hlq*.\$*LPAR_name*\$.*xyzvrm*, where:

- *hlq* is the value of the **Customization library qualifier** field on the **Tools Customizer Settings** panel (CCQPSET)
- *LPAR_name* is the four-character LPAR name
- *xyzvrm* is the three-letter product identifier with the version, release, and modification level

For example, the data set name might be DB2TOOL.PRODUCT.CUST.\$MVS1\$.XYZ410.

WRITE access to this data set is required.

Tools Customizer allocates the data sets for the discover output, the data store, and the customization library with the attributes that are shown in the following table:

| Data set | Organization | Record format | Record length | Block size | Data set name type |
|-------------------------------|---------------------|----------------------|----------------------|-------------------|---------------------------|
| Discover output data set | PO | Variable block | 16383 | 32760 | LIBRARY |
| Data store data set | PO | Variable block | 16383 | 32760 | LIBRARY |
| Product customization library | PO | Fixed block | 80 | 32720 | LIBRARY |

Restrictions:

- Multiple users cannot simultaneously share the discover output data set, data store data set, Tools Customizer metadata library, and metadata library.

Chapter 30. Reference

Reference information supports the tasks that you must complete to install, customize, and use Db2 Cloning Tool.

Cloning scenarios

This topic provides examples of potential cloning scenarios.

These scenarios are intended to be used to assist you in planning your cloning methodology. There are many scenarios that can be achieved. If you do not see a scenario in this topic that meets your requirements, contact IBM Software Support.

Volume cloning using an interim set of volumes

Volumes may be cloned using an interim set of volumes.

Db2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes. The procedure for this type of cloning is:

- The source volumes (referred to as set A) are copied to the interim volumes (set B).
- The interim volumes (set B) are copied to the target volumes (set C).
- The data sets on the target volumes (set C) are renamed.

For this procedure, two COPY commands are used. Each COPY command must be in its own JCL step.

- The first COPY command copies the source volumes to the interim volumes and backs up the source ICF catalogs.
- The second COPY command copies the interim volumes to the target volumes and does not back up the ICF catalogs.

If target ICF catalogs reside on the target volumes (set C) and source ICF catalogs reside on the source volumes (set A), then the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option should be used in the first COPY command, and the BASE-JOURNAL-DDN from the first COPY command should be used in the second COPY command.

If there are no target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option is not used for the first COPY command, then an additional DD must be added to the COPY steps. The additional DD name is VOLPLIST and it must have attributes of RECFM=FB, LRECL=80. The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

The cloning scenario differs depending on whether the target ICF catalogs reside on the target volumes. Each set of COPY command parameters depends on whether the target catalogs are located on the target volumes. If non-target catalogs are placed on target volumes, they should be moved off of the target volumes prior to the volume copy and can be moved back to the target volumes after the RENAME completes.

If no target catalogs are located on the target volumes:

If there are no target ICF catalogs on the target volumes, the following steps are used before the RENAME step.

1. Run COPY of volume set A to volume set B using a USERCATALOGS keyword and including the VOLPLIST DD. The ICF catalog is backed up and the journal file that will be used for the RENAME and Db2 steps will be created. The journal contains volume pairs set A and set B. The VOLPLIST data set created in this run contains the volume pairs from this copy (set A and set B) that will be read in step 3. The TARGET-UCATS-ON-TARGET-VOLUMES(Y) option is not used.

2. Run COPY of volume set B to volume set C using NOUSERCATALOGS keyword and including the VOLPLIST DD. The ICF catalog is not backed up and the journal file that is created in this job will not be further used. The VOLPLIST data set created in this run contains the volume pairs from this copy (set B and set C) and will be read in step 3.
3. Run CKZRNTGT with the VOLPLIST from step 1 on the CKZIN DD and the VOLPLIST from step 2 on the NUCIN DD. Db2 Cloning Tool requires the volume pairs from set A (source) and set C (final target). This step reads the VOLPLIST data set created from the first COPY in step 1 that references set A and set B, and reads the VOLPLIST data set created from the second copy that references set B and set C. It matches up the volume pairs and creates a new temporary data set with the correct volume pairs from set A and set C.
4. Run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN (a temporary data set created in the previous step). This job reads in the temporary data set created in step 3 and is used to update the journal data set created in step 1 to point to volume pairs set A and set C.
5. Run RENAME for volume set C.

Note: Steps 1, 4, and 5 use the same journal data set. Step 2 uses a different journal data set.

If target catalogs are located on the target volumes:

If there are one or more target ICF catalogs on the target volumes, the following steps are used before the RENAME step.

1. Run COPY of volume set A to volume set B using a USERCATALOGS keyword and including TARGET-UCATS-ON-TARGET-VOLUMES(Y). The ICF catalog is backed up and the journal file that will be used for the RENAME and Db2 steps will be created. The journal contains volume pairs set A and set B. The volume pairs from this copy (set A and set B) reside in the journal data set. The volumes of set B in the journal data set will be altered to correspond to the volumes of set C in step 3. The VOLPLIST DD option is not used.
2. Run COPY of volume set B to volume set C using the NOUSERCATALOGS keyword and including the BASE-JOURNAL-DDN keyword pointed to the base journal from step 1. The ICF catalog is not backed up and the journal file created in this job will not be used further. The VOLPLIST DD option is not used. The steps CKZRNTGT and VOLOPTIONS are not needed.
3. Run RENAME for volume set C.

Note: Steps 1 and 3 use the same journal data set. Step 2 uses two journal data sets: the first journal is the journal that was created in step 1 and the second journal is created in step 2.

Example 1: If no target catalogs are located on the target volumes

In the example steps that follow, the following items are used:

- The source volumes are SRC001 and SRC002.
- The interim volumes are INT001 and INT002.
- The target volumes are TGT001 and TGT002

Step 1: Copy SRCxxx to INTxxx and back up the source ICF catalogs

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,
//        DISP=(,CATLG),UNIT=SYSALLDA,
//        RECORG=KS,KEYLEN=64,KEYOFF=0,
//        LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZ.WRK.VOLPLIST,
//        DISP=(,CATLG),UNIT=SYSALLDA,
//        RECFM=FB,LRECL=80,BLKSIZE=0,
```

```

//          SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
FROM-VOLSER(
                SRC001 SRC002
            )
TO-VOLSER(
                INT001 INT002
            )
USERCATALOGS(
                SOURCE.USERCAT1  TARGET.USERCAT1
                SOURCE.USERCAT2  TARGET.USERCAT2
            )
CATWORK-DSN(HLQ?.WRK.* )
JOURNAL-DDN(JOURNAL)
//*

```

Step 2: Copy INTxxx to TGTxxx without backing up the catalog

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZNUC.WRK.VOLPLIST,
//            DISP=(,CATLG),UNIT=SYSALLDA,
//            RECFM=FB,LRECL=80,BLKSIZE=0,
//            SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
FROM-VOLSER(
                INT001 INT002
            )
TO-VOLSER(
                TGT001 TGT002
            )
NOUSERCATALOGS
JOURNAL-DDN(JOURNAL)
//*

```

Step 3: Run CKZRNTGT

Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A
//*   USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE
//*   NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIONS COMMAND
//S2      EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN   DD DSN=HLQ?.CKZ.WRK.VOLPLIST,DISP=SHR
//NUCIN   DD DSN=HLQ?.CKZNUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT  DD DSN=HLQ?.CKZ.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//          DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//

```

Step 4: Run VOLOPTIONS to update journal for RENAME

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=6M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD  DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT  DD  DSN=HLQ?.CKZ.WRK.NEWGT,UNIT=SYSDA,DISP=SHR
//CKZIN   DD  *
          VOLOPTIONS UPDATE           -
          NEWTARGETS-DDN(NEWTGT)      -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 5: RENAME data sets on the TGTxxx volumes

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD  SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD  SYSOUT=*
//JOURNAL DD  DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD  DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD  DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD  *
          RENAME                       -
          SAFE                          -
          VOLBKUP-DDN(VOLBKUP)         -
          RENAME-MASKS(                -
              PROD1.**   TEST1.**      -
              PROD2.**   TEST2.**      -
          )
          JOURNAL-DDN(JOURNAL)
//*
```

Example 2: If target catalogs are located on the target volumes

In the example steps that follow, the following items are used:

- The source volumes are SRC001, SRC002, and SRC003.
- The interim volumes are INT001, INT002, and INT003.
- The target volumes are TGT001, TGT002, and TGT003.
- The source ICF catalog SOURCE.USERCAT1 resides on the source volume SRC001.
- The source ICF catalog SOURCE.USERCAT2 resides on the source volume SRC002.
- The target ICF catalog TARGET.USERCAT1 resides on the target volume TGT001.
- The target ICF catalog TARGET.USERCAT2 resides on the target volume TGT002.

Step 1: Copy SRCxxx to INTxxx and back up the source ICF catalogs

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD  DSN=HLQ?.JRNL,
```

```

//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//CKZIN    DD *
COPY
FROM-VOLSER(
            SRC001 SRC002 SRC003
            )
TO-VOLSER(
            INT001 INT002 INT003
            )
USERCATALOGS(
            SOURCE.USERCAT1 TARGET.USERCAT1
            SOURCE.USERCAT2 TARGET.USERCAT2
            )
TARGET-UCATS-ON-TARGET-VOLUMES(Y)
CATWORK-DSN(HLQ?.WRK.* )
JOURNAL-DDN(JOURNAL)
//*

```

Step 2: Copy INTxxx to TGTxxx without backing up the catalog

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//BASEJRNL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN   DD *
COPY
FROM-VOLSER(
            INT001 INT002 INT003
            )
TO-VOLSER(
            TGT001 TGT002 TGT003
            )
NOUSERCATALOGS
BASE-JOURNAL-DDN(BASEJRNL)
JOURNAL-DDN(JOURNAL)
//*

```

Step 3: RENAME data sets on the TGTxxx volumes

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD *
RENAME
SAFE
VOLBKUP-DDN(VOLBKUP)
RENAME-MASKS(
            PROD1.** TEST1.**
            PROD2.** TEST2.**
            )

```

Volume cloning using multiple interim sets of volumes

Volumes may be cloned using multiple interim sets of volumes.

Db2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes. The procedure for this type of cloning is:

- The source volumes (referred to as set A) are copied to the first set of interim volumes (set B).
- The first set of interim volumes (set B) are copied to the second set of interim volumes (set C).
- The second set of interim volumes (set C) are copied to the target volumes (set D).
- The data sets on the target volumes (set D) are renamed.

For this procedure, four COPY commands are used. Each COPY command must be in its own JCL step.

- The first COPY command copies the source volumes to the first set of interim volumes and backs up the source ICF catalogs.
- The second COPY command copies the first set of interim volumes to the second set of interim volumes, but does not back up the ICF catalogs.
- The third COPY command is used to pair the source volumes to the second interim set of volumes. The ICF catalogs are not backed up.
- The fourth COPY command copies the second set of interim volumes to the target volumes. The ICF catalogs are not backed up.

If target ICF catalogs reside on the target volumes (set C) and source ICF catalogs reside on the source volumes (set A), then the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option should be used in the first COPY command, and the BASE-JOURNAL-DDN from the first COPY command should be used in the second COPY command.

If there are no target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option is not used for the first COPY command, then an additional DD must be added to the COPY steps. The additional DD name is VOLPLIST and it must have attributes of RECFM=FB, LRECL=80. The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

The cloning scenario differs depending on whether the target ICF catalogs reside on the target volumes. Each set of COPY command parameters depends on whether the target catalogs are located on the target volumes. If non-target catalogs are placed on target volumes, they should be moved off of the target volumes prior to the volume copy and can be moved back to the target volumes after the RENAME completes.

If no target catalogs are located on the target volumes:

If there are no target ICF catalogs on the target volumes, the following steps are used before the RENAME step.

1. Run COPY of volume set A to volume set B using the USERCATALOGS keyword and including the VOLPLIST DD. The ICF catalog is backed up and the journal file that will be used for the RENAME and Db2 steps will be created. The journal contains volume pairs set A and set B. The VOLPLIST data set created in this run contains the volume pairs from this copy (set A and set B) that will be read in step 3. The TARGET-UCATS-ON-TARGET-VOLUMES(Y) option is not used.
2. Run COPY of volume set B to volume set C using the NOUSERCATALOGS keyword and including the VOLPLIST DD. The ICF catalog is not backed up and the journal file that is created in this job will not be further used. The VOLPLIST data set created in this run contains the volume pairs from this copy (set B and set C) and will be read in step 3.
3. Run CKZRNTGT with the VOLPLIST from step 1 on the CKZIN DD and the VOLPLIST from step 2 on the NUCIN DD. Db2 Cloning Tool requires the volume pairs from set A (source) and set C (second interim

set of volumes). This step reads the VOLPLIST data set created from the first COPY in step 1 that references set A and set B, and reads the VOLPLIST data set created from the second copy that references set B and set C. It matches up the volume pairs and creates a new temporary data set with the correct volume pairs from set A and set C.

4. Run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN (temporary data set created in previous step). This job reads in the temporary data set created in step 3 and is used to update the journal data set created in step 1 to point to volume pairs set A and set C.
5. Run COPY of volume set A to volume set C using PGM(NONE) and the USERCATALOGS-NOBACKUP keyword, and including the VOLPLIST DD. The ICF catalog is not backed up and the journal file that is created in this job will not be further used. The VOLPLIST data set that is created in this run contains the volume pairs from this copy (set A and set C) and will be read in step 7.
6. Run COPY of volume set C to volume set D using the NOUSERCATALOGS keyword, and including the VOLPLIST DD. The ICF catalog is not backed up and the journal file that is created in this job will not be further used. The VOLPLIST data set created in this run contains the volume pairs from this copy (set C and set D) and will be read in step 7.
7. Run CKZRNTGT with the VOLPLIST created in step 5 on the CKZIN DD and the VOLPLIST from step 6 on the NUCIN DD. Db2 Cloning Tool requires the volume pairs from set A (source) and set D (final target). This step reads the VOLPLIST data set created after the COPY command in step 5 that references set A and set C, and reads the VOLPLIST data set that is created from the second COPY in step 6 that references set C and set D. This step matches up the volume pairs and creates a new temporary data set with the correct volume pairs from set A and set D.
8. Run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN (temporary data set created in previous step). This job reads in the temporary data set created in step 7 and is used to update the journal data set that is created in step 1 and updated in step 4 to point to volume pairs set A and set D.
9. Run RENAME for volume set D.

Note: Steps 1, 4, 8, and 9 use the same journal data set. Steps 2, 5, and 6 use different journal data sets.

This scenario could be extended to additional sets of interim volumes. Each additional set of volumes require the following steps:

1. A COPY job that does not back up the ICF catalogs.
2. A COPY job with PGM(NONE) between the source volumes and each set of interim volumes (starting with the second set).
3. A CKZRNTGT job that includes the VOLPLIST of the preceding pair of COPY jobs.
4. A VOLOPTIONS job to update the journal that was used in the very first COPY job.

If target catalogs are located on the target volumes:

If there are one or more target catalogs on the target volumes, then the following steps are used before the RENAME step.

1. Run COPY of volume set A to volume set B using a USERCATALOGS keyword and TARGET-UCATS-ON-TARGET-VOLUMES(Y). The ICF catalog is backed up and the journal file that will be used for the RENAME and Db2 steps will be created. The journal contains volume pairs set A and set B. The volumes pairs from this copy (set A and set B) reside in the journal data set. The volumes of set B in the journal data set will be altered to correspond to the volumes of set C in step 2. The VOLPLIST DD option is not used.
2. Run COPY of volume set B to volume set C using a NOUSERCATALOGS keyword and including the BASE-JOURNAL-DDN keyword pointing to the base journal from step 1. The ICF catalog is not backed up and the journal file that is created in this job will not be further used. The VOLPLIST DD option is not used, and the steps CKZRNTGT and VOLOPTIONS are not needed.
3. Run COPY of volume set C to volume set D using a NOUSERCATALOGS keyword and including the BASE-JOURNAL-DDN keyword pointing to the base journal from step 1. The ICF catalog is not backed up and the journal file that is created in this job will not be further used. The VOLPLIST DD option is not used, and the steps CKZRNTGT and VOLOPTIONS are not needed.
4. Run RENAME for volume set D.

Note: Steps 1 and 4 use the same journal data set. Steps 2 and 3 each use two journal data sets: the first journal is the journal that was created in step 1, and the second journal is created during its own step (2 or 3).

This scenario could be extended to additional sets of interim volumes. Each additional set of volumes requires a COPY job step that does not back up the ICF catalogs but uses the BASE-JOURNAL-DDN keyword pointed to the base journal from the first COPY command.

Example 1: If no target catalogs are located on the target volumes

In the example steps that follow, the following items are used:

- The source volumes are SRC001 and SRC002.
- The first set of interim volumes are INT101 and INT102.
- The second set of interim volumes are INT201 and INT202.
- The target volumes are TGT001 and TGT002.

Step 1: Copy SRCxxx to INTxxx and back up the source ICF catalogs

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD  DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZ.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN   DD  *
COPY      -
FROM-VOLSER( -
          SRC001 SRC002 -
          ) -
TO-VOLSER( -
          INT101 INT102 -
          ) -
USERCATALOGS( -
          SOURCE.USERCAT1 TARGET.USERCAT1 -
          SOURCE.USERCAT2 TARGET.USERCAT2 -
          ) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*
```

Step 2: Copy INT1xx to INT2xx without backing up the catalog

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD  DSN=HLQ?.NUC1JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZNUC1.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN   DD  *
```

```

COPY -
FROM-VOLSER( -
          INT101 INT102 -
          ) -
TO-VOLSER( -
          INT201 INT202 -
          ) -
NOUSERCATALOGS -
JOURNAL-DDN(JOURNAL) -
//*

```

Step 3: Run CKZRNTGT

Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A
//* USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE
//* NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIIONS COMMAND
//S2 EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN DD DSN=HLQ?.CKZ.WRK.VOLPLIST,DISP=SHR
//NUCIN DD DSN=HLQ?.CKZNUC1.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.CKZ.WRK.NEWTGT1,UNIT=SYSDA,DISP=(,CATLG),
// DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
// SPACE=(CYL,(1,1))

```

Step 4: Run VOLOPTIIONS to update journal for RENAME

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT DD DSN=HLQ?.CKZ.WRK.NEWTGT1,UNIT=SYSDA,DISP=SHR
//CKZIN DD *
VOLOPTIIONS UPDATE -
          NEWTARGETS-DDN(NEWTGT) -
          JOURNAL-DDN(JOURNAL)
//*

```

Step 5: Copy SRCxxx to INT2xx without backing up the catalog to get VOLPLIST

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.NUC2JRNL,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECOGR=KS,KEYLEN=64,KEYOFF=0,
// LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZNUC2.WRK.VOLPLIST,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECFM=FB,LRECL=80,BLKSIZE=0,
// SPACE=(CYL,(1,1))
//CKZIN DD *
COPY -
DATA-MOVER(PGM(NONE)) -

```

```

FROM-VOLSER(          -
        SRC001 SRC002 -
)                    -
TO-VOLSER(           -
        INT201 INT202 -
)                    -
USERCATALOGS-NOBACKUP( -
        SOURCE.USERCAT1 TARGET.USERCAT1 -
        SOURCE.USERCAT2 TARGET.USERCAT2 -
)                    -
CATWORK-DSN(HLQ?.WRK2.* ) -
JOURNAL-DDN(JOURNAL)  -
//*

```

Step 6: Copy INT2xx to TGTxxx without backing up the catalog

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD  DSN=HLQ?.NUC3JRNL,
//        DISP=(,CATLG),UNIT=SYSALLDA,
//        RECOG=KS,KEYLEN=64,KEYOFF=0,
//        LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZNUC3.WRK.VOLPLIST,
//        DISP=(,CATLG),UNIT=SYSALLDA,
//        RECFM=FB,LRECL=80,BLKSIZE=0,
//        SPACE=(CYL,(1,1))
//CKZIN   DD  *
COPY      -
FROM-VOLSER( -
        INT201 INT202 -
)        -
TO-VOLSER( -
        TGT001 TGT002 -
)        -
NOUSERCATALOGS -
JOURNAL-DDN(JOURNAL)
//*

```

Step 7: Run CKZRNTGT

Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A
//* USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE
//* NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIIONS COMMAND
//S2      EXEC  PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD  DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD  DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN   DD  DSN=HLQ?.CKZNUC2.WRK.VOLPLIST,DISP=SHR
//NUCIN    DD  DSN=HLQ?.CKZNUC3.WRK.VOLPLIST,DISP=SHR
//NEWTGT  DD  DSN=HLQ?.CKZ.WRK.NEWTGT2,UNIT=SYSDA,DISP=(,CATLG),
//        DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//        SPACE=(CYL,(1,1))

```

Step 8: Run VOLOPTIIONS to update journal for RENAME

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT  DD DSN=HLQ?.CKZ.WRK.NEWTGT2,UNIT=SYSDA,DISP=SHR
//CKZIN   DD *
          VOLOPTIONS UPDATE      -
          NEWTARGETS-DDN(NEWTGT) -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 9: RENAME data sets on the TGTxxx volumes

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD *
          RENAME                -
          SAFE                   -
          VOLBKUP-DDN(VOLBKUP)   -
          RENAME-MASKS(         -
              PROD1.** TEST1.** -
              PROD2.** TEST2.** -
          )                      -
          JOURNAL-DDN(JOURNAL)
//*
```

Example 2: If target catalogs are located on the target volumes

In the example steps that follow, the following items are used:

- The source volumes are SRC001, SRC002, and SRC003.
- The interim volumes are INT101, INT102, and INT103.
- The interim volumes are INT201, INT202, and INT203.
- The target volumes are TGT001, TGT002, and TGT003.
- The source ICF catalog SOURCE.USERCAT1 resides on the source volume SRC001.
- The source ICF catalog SOURCE.USERCAT2 resides on the source volume SRC002.
- The target ICF catalog TARGET.USERCAT1 resides on the target volume TGT001.
- The target ICF catalog TARGET.USERCAT2 resides on the target volume TGT002.

Step 1: Copy SRCxxx to INTxxx and back up the source ICF catalogs

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
```

```

//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//CKZIN DD *
COPY -
FROM-VOLSER( -
          SRC001 SRC002 SRC003 -
          ) -
TO-VOLSER( -
          INT101 INT102 INT103 -
          ) -
USERCATALOGS( -
          SOURCE.USERCAT1 TARGET.USERCAT1 -
          SOURCE.USERCAT2 TARGET.USERCAT2 -
          ) -
TARGET-UCATS-ON-TARGET-VOLUMES(Y) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL) -
//*

```

Step 2: Copy INT1xx to INT2xx without backing up the catalog

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//BASEJRNL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN DD *
COPY -
FROM-VOLSER( -
          INT101 INT102 INT103 -
          ) -
TO-VOLSER( -
          INT201 INT202 INT203 -
          ) -
NOUSERCATALOGS -
BASE-JOURNAL-DDN(BASEJRNL) -
JOURNAL-DDN(JOURNAL) -
//*

```

Step 3: Copy INT2xx to TGTxxx without backing up the catalog

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//BASEJRNL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN DD *
COPY -
FROM-VOLSER( -
          INT201 INT202 INT203 -
          ) -
TO-VOLSER( -
          TGT001 TGT002 TGT003 -
          ) -
NOUSERCATALOGS -
BASE-JOURNAL-DDN(BASEJRNL) -

```

```
JOURNAL-DDN(JOURNAL)
//*
```

Step 4: RENAME data sets on the TGTxxx volumes

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD *
          RENAME                -
          SAFE                   -
          VOLBKUP-DDN(VOLBKUP)   -
          RENAME-MASKS(         -
              PROD1.** TEST1.** -
              PROD2.** TEST2.** -
          )                       -
          JOURNAL-DDN(JOURNAL)
//*
```

Db2 subsystem cloning using data set copy

Db2 subsystems can be cloned using data set copy. The source data sets can be copied and renamed to target data set names. The target data sets can then be conditioned and used for the target Db2 system.

Overview

Subsystem cloning using data set copy provides an alternative to using volume copy. Volume copy and rename can be faster than copying by data set, but there might be cases where it is not feasible to clone by volume.

To clone a Db2 subsystem using data set copy, the data sets must all be at a consistent point in time. Currently, the only way for the target data sets to be at a consistent point in time is for the source Db2 system to be down while the data set copies are being processed.

Cloning Db2 subsystems by data set copy allows the source Db2 system to reside in a mixed volume environment, where multiple Db2 systems or other application data reside on the same set of volumes. The target data sets can then reside in a mixed volume environment or they can reside on dedicated volumes.

The following procedure allows for repetitive cloning.

- The source data sets (referred to as set A) are copied and renamed to target data sets (referred to as set B) by a Db2 Cloning Tool COPY-BY-DS command.
- The target Db2 subsystems contained in the target data sets are conditioned. The Db2 conditioning updates the appropriate Db2 data sets in the target data sets (set B) to the new data set names and volume serials.

The COPY-BY-DS command uses the RENAME-MASKS keyword to identify the source data sets and how they should be renamed to target names. The command creates a journal data set, which is used throughout the rename and Db2 conditioning steps.

If the source volumes contain data sets that are encrypted with DFSMS pervasive encryption, ensure the following:

- For cross-LPAR cloning, any key labels that are used to protect source Db2 data sets must be defined on the target LPAR and must refer to the same encryption key.
- The target Db2 started task user ID is permitted to use any key labels that are used to protect Db2 data sets that are cloned from the source Db2 subsystem.
- If target Db2 data sets already exist, they must have the same encryption status as their counterparts on the source Db2. This is an ADRDSSU COPY utility requirement. To ensure that an encryption status mismatch does not lead to copy errors, you can manually delete all target Db2 data sets before the cloning.

Step overview

This section summarizes the steps you need to follow to perform Db2 subsystem cloning using data set copy.

1. Run DB2STOP to stop the source Db2 system.
2. Run COPY-BY-DS. Use RENAME-MASKS to specify the source data sets (set A) to copy and rename to the target names (set B).
3. Run DB2START to start the source Db2 system.
4. Run the Db2 conditioning commands: DB2UPDATE, DB2START, DB2SQL, DB2STOP.

Example

In the example steps that follow, the following items are used:

- The source Db2 (DB2P) resides in the source data sets (DB2P.**). The source data sets consist of the BSDS (DB2P.BSDS0x), active logs (DB2P.LOGx), and the Db2 table and index spaces (DB2P.DSNDBC.**). The source archive log data sets (DB2P.ARCHLOGx.***) are not copied as part of the cloning.
- The target Db2 (DB2T) will reside in the target data sets (DB2T.**).

Step 1

The first step is to stop the source Db2 system. Sample JCL can be found in the installation SCKZJCL library in member CKZDSTO.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CKZIN   DD *
          DB2STOP
          DB2-SSID(DB2P)
//*
```

Step 2

The second step is to copy the source data sets and rename to target data set names. The RENAME-MASKS and EXCLUDE-SRCNAME-MASKS keywords expect the source data set component names and not the cluster names. Sample JCL can be found in the installation SCKZJCL library in member CKZCPYDS.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
```

```

//CKZIN DD *
COPY-BY-DS -
OFFLINE -
DATA-MOVER( PGM(ADRSSU) -
FASTREP(PREF) -
DSNS-PER-COPY(255) -
DSS-COPY-COMMANDS(5) -
) -
REPLACE-UNCONDITIONAL -
RENAME-MASKs( -
DB2P.BSDS0%.DATA DB2T.BSDS0%.DATA -
DB2P.LOG%.DATA DB2T.LOG%.DATA -
DB2P.DSNDBD.** DB2T.DSNDBD.** -
) -
EXCLUDE-SRCNAME-MASKS( -
DB2P.ARCHLOG%.** -
) -
JOURNAL-DDN(JOURNAL) -
//*

```

Step 3

The third step is to start the source Db2 system. Sample JCL can be found in the installation SCKZJCL library in member CKZDSTA.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CKZIN DD *
DB2START -
DB2-SSID(DB2P)
//*

```

Step 4

The last step is to run the Db2 conditioning commands. These commands are:

- DB2UPDATE
- DB2START
- DB2SQL
- DB2STOP

For additional information about the conditioning commands, refer to the appropriate section in [“Db2 offline cloning”](#) on page 126 for your specific type of cloning.

Db2 subsystem cloning from a Db2 BACKUP SYSTEM backup

Db2 subsystems can be cloned using the backup volumes created by a Db2 BACKUP SYSTEM command.

Overview

Because the source Db2 is running at the time of the Db2 BACKUP SYSTEM command, this is an online cloning. This situation is similar to cloning using an interim set of volumes. Db2 Cloning Tool supports this process but requires knowledge of the original source and final target volumes.

If the target volumes are set up in storage groups in a way that mirrors the source volume configuration, it is possible to pair the backup volumes to the target volumes by using the source and target storage group names. For example, if the source has two storage groups, one for logs (SRCSGL) and one for data (SRCSGD), and the target has two storage groups, one for logs (TGTSGL) and one for data (TGTSGD), then the backup volumes can be paired to the target volumes by using the source and target storage group names. The backup volumes that correspond to the source log storage group (SRCSGL) are paired with

volumes in the target log storage group (TGTSGL), and the backup volumes that correspond to the source data storage group (SRCSGD) are paired with volumes in the target data storage group (TGTSGD).

The following procedure allows for repetitive cloning without requiring a manual update of the volume specifications when the LAST keyword is used with the DB2GETBACKINFO command.

- The source volumes (referred to as set A) have been copied to the backup volumes (set B) by a Db2 BACKUP SYSTEM command.
- The backup volumes (set B) are copied to the target volumes (set C).
- The data sets on the target volumes (set C) are renamed.
- The target Db2 subsystems on the target volumes are conditioned.
- Db2 Cloning Tool requires the original source volumes (set A) and the final target volumes (set C) to rename and catalog the target volume data sets and to update the target Db2 system to replace the original source volumes with the final target volumes. To accomplish this, Db2 Cloning Tool must know the pairing between set A and set B and then pair those volumes from set A to the final target volumes (set C).
- To obtain the original source volumes (set A) and the backup volumes (set B) from DFSMSshsm, the DB2GETBACKINFO command is executed to query DFSMSshsm. Any generation can be selected in the DB2GETBACKINFO command. In this example, LAST is used for the most current version pairs of set A and set B. This list is provided in DFSMSshsm LIST format in HSMLIST DD. Db2 Cloning Tool also creates the BACKINFO DD data set in a format that is consistent with a utility in Db2 Recovery Expert for z/OS and Rocket Mainstar Database Backup and Recovery for Db2 on z/OS (DBR for Db2), only for the purpose of keeping cloning from a system backup consistent across products. The HSMLIST DD is not used again. The source ICF catalogs must also be identified by the user in this step.
- The job that executes the BACKINFO-REFORMAT command does several things:
 - Reads the BACKINFO DD data set created in the previous step and creates the VOLPAIRS DD data set with volume pairs set A and set B. These files will be used in subsequent COPY steps.
 - Creates the USERSGDEFS-DDN(USRSGDEF) with the set B volumes and the storage group name of their corresponding source volume, which becomes the input to the clone in the second COPY step.
 - The user catalogs are added to this step and stored in the data set pointed to by UCATS DDN so that they can later be passed to the first COPY command.

For this procedure, two COPY commands are used. Each COPY command must be in its own JCL step.

The cloning scenario differs somewhat, depending on whether target ICF catalogs reside on the target volumes. Each set of COPY command parameters depends on whether target catalogs are located on the target volumes. If non-target catalogs are placed on target volumes, they should be moved off of the target volumes prior to the volume copy, and can be moved back to the target volumes after the RENAME completes.

If no target catalogs are located on the target volumes

If there are no target catalogs on the target volumes, the following steps are used before the RENAME step.

The first COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword and does NOT do any volume copies as the copies (set A to set B) were done by the Db2 BACKUP SYSTEM command.
- TARGET-UCATS-ON-TARGET-VOLUMES(Y) is not used.
- Identifies the source volume (set A) to backup volume (set B) pairing by reading the VOLPAIRS-DDN created in the previous step and creates a new data set pointed to by the VOLPLIST DD HLQ?.WRK.VOLPLIST. Another VOLPLIST is created in the second COPY. These will be used later.
- Backs up the source ICF catalog copies from the BACKUP SYSTEM backup volumes.
- Creates a journal data set, which is used throughout the RENAME and Db2 conditioning steps.

The second COPY command:

- Copies the backup volumes (set B) pointed to by FROM-USER-STORAGEGROUP and USERSGDEFS-DDN(USRSGDEF) to the target volumes (set C) pointed to by TO-STORAGEGROUP. KEEP-VOLUMES-SEQUENCE(Y) should be specified to copy from SRCSGL to TGTSGL and SRCSGD to TGTSGD storage groups, respectively.
- Volume sets (set B) and (set C) volume pairs are added to a new VOLPLIST data set, HLQ?.NUC.WRK.VOLPLIST.
- The source ICF catalogs are NOT backed up again.
- A journal data set is created but will NOT be used again. The journal from the first copy will be used.

The two VOLPLIST data sets created in each of the copy commands which contain set A to set B and set B to set C (pointed to by DDs CKZIN and NUCIN) are read to create a data set pointed to by DD NEWTGT. This data set contains the pairing set A to set B to set C.

After the COPY steps, the data set that is pointed to by DD NEWTGT is read and the journal that created in the first COPY step is updated with the volumes from set A to set C, so the data sets can be renamed and Db2 can be conditioned appropriately.

If target catalogs are located on the target volumes

If there are one or more target catalogs on the target volumes, then the following steps are used before the RENAME step. The first COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword and does NOT do any volume copies as the copies (set A to set B) were done by the Db2 BACKUP SYSTEM command.
- TARGET-UCATS-ON-TARGET-VOLUMES(Y) is used.
- Identifies the source volume (set A) to backup volume (set B) pairing by reading the VOLPAIRS-DDN created in the previous step. A VOLPLIST DD is not created.
- Backs up the source ICF catalog copies from the BACKUP SYSTEM backup volumes.
- Creates a base journal data set which is used for the source to backup volume pairing. This journal contains information about the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter, which is required for the second COPY command.

The second COPY command:

- Does not use VOLPLIST data sets. The BASE-JOURNAL-DDN points to the base journal from the first COPY command. The base journal is updated with the source-to-target volume pairing and the source-to-target ICF catalog pairing.
- Issues CAS UNALLOCATE commands for the target ICF catalogs that reside on the target volumes.
- Copies the backup volumes (set B) pointed to by FROM-USER-STORAGEGROUP and USERSGDEFS-DDN(USRSGDEF) to the target volumes (set C) pointed to by TO-STORAGEGROUP. KEEP-VOLUMES-SEQUENCE(Y) should be specified to copy from SRCSGL to TGTSGL and SRCSGD to TGTSGD storage groups, respectively.
- Does NOT back up the source ICF catalogs again.
- Creates a new journal data set, but that journal data set will NOT be used again. The base journal from the first COPY will be used.

After the COPY steps

The RENAME renames and catalogs all of the target volume data sets (set C) to new names.

The Db2 conditioning updates the appropriate Db2 data sets on the target volumes (set C) to the new data set names and volume serials.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the Db2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the

DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

For data sharing, the first members started will receive a DSNR020I WTOR due to the conditional restart records contained in the other members. If the start jobs are being run serially by a job scheduler, it is recommended that the Db2START command for all but the last member include the STOP-WAITING-IF-DSNR020I keyword with a value of Y. This will allow the starts of the other members to be done concurrently. After the last member has been started, a DB2START for the first member should be done again using the WAITONLY and DB2-ALREADY-RUNNING(RC(0)) keywords. Completion of the DB2START with a return code of 4 or less indicates that the first member is up and ready for the remainder of the Db2 conditioning jobs.

Step overview

This section summarizes the steps you need to follow to perform Db2 subsystem cloning.

1. Run DB2GETBACKINFO to get the source (set A) and backup (set B) volume pairing used by the Db2 BACKUP SYSTEM command. (Any available generation can be specified; our example uses LAST.)
2. Run BACKINFO-REFORMAT to take the backinfo data set created by the DB2GETBACKINFO command (Step 1) and reformat it for use by subsequent COPY commands (Steps 3 and 4).
3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a USERCATALOGS-DDN keyword (data set from step 2), and include the VOLPLIST DD. If there are target volumes that have target ICF catalogs on them, then TARGET-UCATS-ON-TARGET-VOLUMES(Y) must be specified, and VOLPLIST DD should not be used. Otherwise, TARGET-UCATS-ON-TARGET-VOLUMES should be set to N (or should not be specified) and VOLPLIST DD must be specified.

The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

4. Run COPY using FROM-USER-STORAGEGROUP and USERSGDEFS-DDN to get the backup volumes (set B) (data set from step 2), using the NOUSERCATALOGS keyword and including the VOLPLIST DD.
 - For strict copy from volumes of the source storage group to volumes of the target storage group (for data or logs), the KEEP-VOLUMES-SEQUENCE(Y) parameter should be used.
 - If there are target volumes that have target ICF catalogs on them, then BASE-JOURNAL-DDN must be specified and must point to the journal from the first COPY command, and the VOLPLIST DD should not be specified. Otherwise, specify the VOLPLIST DD, and do not specify BASE-JOURNAL-DDN.

The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

5. If there are no target ICF catalogs on the target volumes, run CKZRNTGT with the VOLPLIST from step 3 on the CKZIN DD and the VOLPLIST from step 4 on the NUCIN DD. If there are target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) was used in step 3, omit this step.
6. If there are no target ICF catalogs on the target volumes, run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN. If there are target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) was used in step 3, omit this step.
7. Run RENAME for target volumes (set C).
8. Run the Db2 conditioning commands: DB2UPDATE, DB2ALTERBSDS(SLB-START), DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP, DB2UTILXCLEAN.

Note: Steps 3, 6, 7, and 8 use the same journal data set. If there are target ICF catalogs on target volumes, then step 4 uses two journal data sets: the first gathered from step 3 and the second created in step 4. If there are no target ICF catalogs on the target volumes, then step 4 uses a different journal data set.

Example

In the example steps that follow, the following items are used:

- The source Db2 resides on the source volumes (SRCxxx). The source log volumes are in storage group SRCGL and the source data volumes are in storage group SRCSGD.
- The target Db2 will reside on the target volumes (TGTxxx). The target log volumes are in storage group TGTGL and the target data volumes are in storage group TGTSGD.
- There are one or more sets of backup volumes (BKPxxx) that have been created by Db2 BACKUP SYSTEM commands.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.
- There are two target ICF catalogs (USERCAT.TGT01 and USERCAT.TGT02). It is possible that one of the ICF catalogs resides on a target volume, both ICF catalogs reside on target volumes, or none of the target volumes contain ICF catalogs.

Step 1

The first step is to get the source to backup volume pairing (SRCxxx to BKPxxx) and identify the source ICF catalogs from the last Db2 BACKUP SYSTEM taken for location DB2PLOC. Sample JCL can be found in the installation SCKZJCL library in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DSN=HLQ?.WRK.BACKINFO,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD  DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD  *
DB2GETBACKINFO          -
BACKINFO-DDN(BACKINFO) -
WORK-DDN(HSMLIST)      -
LAST                   -
LOCATION(DB2PLOC)       -
USERCATALOGS(         -
  USERCAT.SRC01       -
  USERCAT.SRC02       -
)
//*
```

Step 2

The second step is to reformat the output of step 1 (backinfo data set) for use in the COPY in steps 3 and 4. The user catalog pairs are also specified here. Sample JCL can be found in the installation SCKZJCL library in member CKZBKIRF.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//USRSGDEF DD DSN=HLQ?.WRK.USRSGDEF,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD  DSN=HLQ?.WRK.UCATS,
```

```

//          DISP=( ,CATLG ),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
BACKINFO-REFORMAT          -
BACKINFO-DDN(BACKINFO)    -
VOLPAIRS-DDN(VOLPAIRS)    -
USERSGDEFS-DDN(USERSGDEF) -
USERCATALOGS-DDN(UCATS)   -
USERCATALOGS(             -
  USERCAT.SRC01 USERCAT.TGT01 -
  USERCAT.SRC02 USERCAT.TGT02 -
)
//*
```

Step 3

The third step is to set the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal and back up the source ICF catalogs from the backup volumes. Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step when there are target ICF catalog on the target volumes follows:

```

//S1      EXEC PGM=00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=( ,CATLG ),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//CKZIN   DD *
COPY          -
DATA-MOVER(PGM(NONE)) -
VOLPAIRS-DDN(VOLPAIRS) -
USERCATALOGS-DDN(UCATS) -
TARGET-UCATS-ON-TARGET-VOLUMES(Y) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*
```

Partial JCL for this step when there are no target ICF catalog on the target volumes follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=( ,CATLG ),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//          DISP=( ,CATLG ),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY          -
DATA-MOVER(PGM(NONE)) -
VOLPAIRS-DDN(VOLPAIRS) -
USERCATALOGS-DDN(UCATS) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*
```

Step 4

The fourth step is to copy the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs. Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step when there are target ICF catalog on the target volumes follows:

```
//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//BASEJRNL DD DISP=SHR,DSN=DSN=HLQ?.JRNL
//CKZIN DD *
COPY
  FROM-USER-STORAGEGROUP(
                        SRCUGL SRCUGD
                        )
  USRSGDEFS-DDN(USRSGDEF)
  USRSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLXCL(18))
  TO-STORAGEGROUP(
                TGUGL TGUGD
                )
  KEEP-VOLUMES-SEQUENCE(YES)
  NOUSERCATALOGS
  BASE-JOURNAL-DDN(BASEJRNL)
  JOURNAL-DDN(JOURNAL)
//*
```

Partial JCL for this step when there are no target ICF catalog on the target volumes follows:

```
//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
//           DISP=(,CATLG),UNIT=SYSALLDA,
//           RECFM=FB,LRECL=80,BLKSIZE=0,
//           SPACE=(CYL,(1,1))
//CKZIN DD *
COPY
  FROM-USER-STORAGEGROUP(
                        SRCUGL SRCUGD
                        )
  USRSGDEFS-DDN(USRSGDEF)
  USRSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLXCL(18))
  TO-STORAGEGROUP(
                TGUGL TGUGD
                )
  KEEP-VOLUMES-SEQUENCE(YES)
  NOUSERCATALOGS
  JOURNAL-DDN(JOURNAL)
//*
```

Step 5

The fifth step is to run CKZRNTGT with the VOLPLIST data sets from steps 3 and 4. Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 3), this step is omitted.

Partial JCL for this step follows:

```
//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A
//* USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE
//* NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIONS COMMAND
//S2 EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
```

```
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
// DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
// SPACE=(CYL,(1,1))
```

Step 6

This is the sixth step of run VOLOPTIONS with the newtgt data set from step 5 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME. Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 3), this step is omitted.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT DD DISP=SHR,DSN=HLQ?.WRK.NEWTGT
//CKZIN DD *
    VOLOPTIONS UPDATE -
    NEWTARGETS-DDN(NEWTGT) -
    JOURNAL-DDN(JOURNAL)
//*
```

Step 7

The seventh step of the RENAME the data sets on the target volumes (TGTxxx). Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL and the RENAME command for this step follow:

```
//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
// UNIT=SYSALLDA,DISP=(,CATLG),
// SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
// UNIT=SYSALLDA,DISP=(,CATLG),
// SPACE=(CYL,(40,40))
//CKZIN DD *
    RENAME -
    SAFE -
    VOLBKUP-DDN(VOLBKUP) -
    RENAME-MASKS(
        PROD1.** TEST1.** -
        PROD2.** TEST2.** -
    ) -
    JOURNAL-DDN(JOURNAL)
//*
```

Step 8

The last step is to run the Db2 conditioning commands. These commands are:

- DB2UPDATE
- DB2ALTERBSDS SLB-START
- DB2START SPECIAL

- DB2FIX DATABASES(DB2)
- DB2SQL
- DB2FIX DATABASES(APPLICATION)
- DB2SCHEMA-UPDATE
- DB2STOP
- DB2UTILXCLEAN

For additional information about the conditioning commands, refer to the appropriate section in “[Db2 online cloning](#)” on page 134 for your specific type of cloning.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the Db2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

The following are examples of the DB2UPDATE, DB2ALTERBSDS, and DB2START steps necessary to handle this condition.

Example 1 – a non-data sharing subsystem with active logs that are striped:

1. Update the Db2 directory and the BSDS.

```
DB2UPDATE          -
  DB2-HLQS( srcvcat tgtvcat ) -
  DDF( ... )       -
  JOURNAL-DDN(JOURNAL)
```

2. Create a system-level backup (SLB) start conditional restart record in the BSDS.

```
DB2ALTERBSDS      -
  SLB-START        -
  JOURNAL-DDN(JOURNAL)
```

3. Start the target Db2 in special mode and automatically reply to the Db2 restart WTOR.

```
DB2START          -
  DB2-SSID(ssid)   -
  SPECIAL          -
  DSNZPARM(ssidSPEC) -
  REPLY-TO-RESTART-WTOR(Y)
```

Example 2 – a data sharing group with three members (the active logs may or may not be striped):

1. Update the Db2 directory and the BSDS for mbr1.

```
DB2UPDATE          -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1 -
    srcmbr2 mbr2           -
    srcmbr3 mbr3 )        -
  DDF( ... )           -
  JOURNAL-DDN(JOURNAL)
```

2. Update the BSDS for mbr2.

```
DB2UPDATE          -
  BSDSONLY         -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1 -
    srcmbr2 mbr2           -
    srcmbr3 mbr3 )        -
  DDF( ... )           -
  JOURNAL-DDN(JOURNAL)
```

3. Update the BSDS for mbr3.

```

DB2UPDATE                -
  BSDSONLY                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1 -
    srcmbr2 mbr2          -
    srcmbr3 mbr3 )       -
  DDF( ... )             -
  JOURNAL-DDN(JOURNAL)

```

4. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr1.

```

DB2ALTERBSDS            -
  DB2-MEMBER(mbr1)     -
  SLB-START             -
  JOURNAL-DDN(JOURNAL)

```

5. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr2.

```

DB2ALTERBSDS            -
  DB2-MEMBER(mbr2)     -
  SLB-START             -
  JOURNAL-DDN(JOURNAL)

```

6. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr3.

```

DB2ALTERBSDS            -
  DB2-MEMBER(mbr3)     -
  SLB-START             -
  JOURNAL-DDN(JOURNAL)

```

7. Start the target Db2 mbr1 in special mode, automatically reply to the Db2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

DB2START                -
  DB2-SSID(mbr1)        -
  SPECIAL                -
  DSNZPARM(mbr1SPEC)    -
  REPLY-TO-RESTART-WTOR(Y) -
  STOP-WAITING-IF-DSNR020I(Y)

```

8. Start the target Db2 mbr2 in special mode, automatically reply to the Db2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

DB2START                -
  DB2-SSID(mbr2)        -
  SPECIAL                -
  DSNZPARM(mbr2SPEC)    -
  REPLY-TO-RESTART-WTOR(Y) -
  STOP-WAITING-IF-DSNR020I(Y,RC(3))

```

9. Start the target Db2 mbr3 in special mode and automatically reply to the Db2 restart WTOR.

```

DB2START                -
  DB2-SSID(mbr3)        -
  SPECIAL                -
  DSNZPARM(mbr3SPEC)    -
  REPLY-TO-RESTART-WTOR(Y)

```

10. Stop the target Db2 mbr3.

```

DB2STOP                 -
  DB2-SSID(mbr3)

```

11. Stop the target Db2 mbr2.

```

DB2STOP                 -
  DB2-SSID(mbr2)

```

12. Wait for the target Db2 mbr1 to complete its startup.

```
DB2START -
DB2-SSID(mbr1) -
SPECIAL -
DSNZPARM(mbr1SPEC) -
REPLY-TO-RESTART-WTOR(Y) -
DB2-ALREADY-RUNNING(RC(0)) -
WAITONLY
```

Db2 subsystem cloning to a specific point in time from a Db2 BACKUP SYSTEM backup

Db2 subsystems can be cloned by using the backup volumes that are created by a Db2 BACKUP SYSTEM command. As of Db2 V11, it is possible to clone to a specific point in time that is different from the point in time when the BACKUP SYSTEM was taken. This cloning scenario uses the RESTORE SYSTEM utility run using LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*). The point in time to use is set by using the DSNJU003 (change log inventory) utility to create a conditional restart record using SYSPITR or SYSPITRT.

Overview

Because the source Db2 is running at the time of the Db2 BACKUP SYSTEM command, this is an online cloning. This situation is similar to cloning by using an interim set of volumes. Db2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes.

For this scenario, there must be at least two Db2 system level backups (SLBs) for the Db2 system that is being cloned. The cloning will be to a specific point in time that lies between two of the SLBs. For example, there are SLBs at points-in-time 1 and 2, and the cloning will be to point-in-time 1.5. The cloning uses the data volumes from point-in-time 1 and the log volumes from point-in-time 2. The RESTORE SYSTEM utility is used to apply the logs up to the required point in time 1.5.

This scenario references volume sets in the following manner:

Set A

The source volumes.

Set B

The system level backup volume. Subsets of these volumes are:

- Set B1 is the SLB volume set for point-in-time 1.
- Set B2 is the SLB volume set for point-in-time 2.

Set C

Target volumes.

Data (D) or Log (L) volumes

To specifically identify the data and log volumes from each set, a D (for data volumes) or an L (for log volumes) is appended the volume set name. For example: B1D is the backup data volume set for point-in-time 1.

- The source volumes (set A) already have been copied to the backup volumes (sets B1 and B2) by Db2 BACKUP SYSTEM commands.
- The backup data volumes for point-in-time 1 (set B1D) are copied to the target data volumes (set CD) and the backup log volumes for point-in-time 2 (set B2L) are copied to the target log volumes (set CL).
- The data sets on the target volumes (set C) are renamed.
- The target Db2 subsystems BSDS are conditioned.
- The conditional restart records using SYSPITR or SYSPITRT are created.
- The RESTORE SYSTEM utility is run with LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*).
- The target Db2 subsystems on the target volumes are conditioned.

- Db2 Cloning Tool requires the original source volumes (set A) and the final target volumes (set C) to rename and catalog the target volume data sets, and to update the target Db2 system to replace the original source volumes with the final target volumes. To accomplish this, Db2 Cloning Tool must know the pairing between set A and set B1 and then pair those volumes from set A to the final target volumes (set C).
- To obtain the original source volumes (set A) and the backup volumes for point-in-time 1 (set B1) and point-in-time 2 (set B2) from DFSMSHsm, the DB2GETBACKINFO command is executed to query DFSMSHsm. Any generation can be selected in the DB2GETBACKINFO command. In this example, the token that applies to the point-in-time 1 SLB is used to get the list of data volume pairs of set A and set B1, and the token that applies to the point-in-time 2 SLB is used to get the list of log volume pairs of set A and set B2. The lists are provided in DFSMSHsm LIST format in HSMLIST DD. Db2 Cloning Tool also creates the BACKINFO DD data set in a format that is consistent with a utility in Db2 Recovery Expert and Rocket Mainstar DBR for Db2 on z/OS, only for the purpose of keeping cloning from a system backup consistent across products. The HSMLIST DD is not used again. The source ICF catalogs must also be identified by the user in this step.
- The job that executes the BACKINFO-REFORMAT command:
 - Reads the BACKINFO DD data sets created in the previous step and creates the VOLPAIRS DD data set with volume pairs set A and set B. These files are used in subsequent COPY steps.
 - Creates the USERSGDEFS-DDN(USRSGDEF) with the set B volumes and the storage group name of their corresponding source volume, which becomes the input to the clone in the second COPY step.
 - The user catalogs are added to this step and stored in the data set pointed to by UCATS DDN so that they can later be passed to the first COPY command.

For this part of the procedure, two COPY commands are used. Each COPY command must be in its own JCL step.

Each set of COPY command parameters depends on whether the target catalogs are located on the target volumes. If non-target catalogs are placed on target volumes, they should be moved off of the target volumes prior to the volume copy and can be moved back to the target volumes after the RENAME completes.

If no target catalogs are located on the target volumes

If there are no target catalogs on the target volumes, the following steps are used before the RENAME step.

The first COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword and does NOT do any volume copies, as the copies (set A to set B) were done by the Db2 BACKUP SYSTEM commands.
- TARGET-UCATS-ON-TARGET-VOLUMES(Y) is not used.
- Identifies the source volume (set A) to backup volume (set B) pairing by reading the VOLPAIRS-DDN created in the previous step and creates a new data set pointed to by the VOLPLIST DD HLQ?.WRK.VOLPLIST. Another VOLPLIST is created in the second COPY. These are used later.
- Backs up the source ICF catalog copies from the BACKUP SYSTEM backup volumes.
- Creates a journal data set, which is the journal data set that is used throughout the RENAME and Db2 conditioning steps.

The second COPY command:

- Copies the backup volumes (set B) contained in the DD pointed to by the FROM-USER-STORAGEGROUP and USERSGDEFS-DDN(USRSGDEF) to the target volumes (set C) pointed to by the TO-STORAGEGROUP.
- Volume sets (set B) and (set C) volume pairs are added to a new VOLPLIST data set, HLQ?.NUC.WRK.VOLPLIST.
- The source ICF catalogs are NOT backed up again.

- A journal data set is created, but will not be used again. The journal from the first copy is used.

The two VOLPLIST data sets created in each of the COPY commands, which contain set A to set B and set B to set C (pointed to by DDs CKZIN and NUCIN), are read to create a data set pointed to by DD NEWTGT. This data set contains the pairing set A to set B to set C.

Next, the data set pointed to by DD NEWTGT is read and the journal that was created in the first COPY command is updated with the volumes from set A to set C, so the data sets can be renamed and Db2 can be conditioned appropriately.

If target catalogs are located on the target volumes

If there are one or more target catalogs on the target volumes, the following steps are used before the RENAME step.

The first COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword and does not do any volume copies, as the copies (set A to set B) were done by the Db2 BACKUP SYSTEM command.
- TARGET-UCATS-ON-TARGET-VOLUMES(Y) is used.
- Identifies the source volume (set A) to the backup volume (set B) pairing by reading the VOLPAIRS-DDN created in the previous step. A VOLPLIST DD is not created.
- Backs up the source ICF catalog copies from the BACKUP SYSTEM backup volumes.
- Creates a base journal data set, which is used for the source to backup volume pairing. This journal contains information about the TARGET-UCATS-ONTARGET-VOLUMES(Y) parameter, which is required for the second COPY command.

The second COPY command:

- Does not use VOLPLIST data sets. The BASE-JOURNAL-DDN points to the base journal from the first COPY command. The base journal is updated with the source-to-target volume pairing and the source-to-target ICF catalog pairing.
- Issues CAS UNALLOCATE commands for the target ICF catalogs that reside on the target volumes.
- Copies the backup volumes (set B) pointed to by FROM-USERSTORAGEGROUP and USERSGDEFS-DDN(USRSGDEF) to the target volumes (set C) pointed to by TO-STORAGEGROUP. KEEP-VOLUMES-SEQUENCE(Y) should be specified to copy from SRCSGL to TGTSG and SRCSGD to TGTSGD storage groups, respectively.
- Does not back up the source ICF catalogs again.
- Creates a new journal data set, but that journal data set will not be used again. The base journal from the first COPY will be used.

After the COPY steps

The RENAME renames and catalogs all of the target volume data sets (set C) to new names.

The Db2 BSDS conditioning updates the Db2 BSDS and directory data sets on the target volumes (set C) to the new data set names and volume serials and creates the SYSVALUE data set that will be used by the RESTORE SYSTEM utility.

The conditional restart records are created for the required point in time using SYSPITR or SYSPITRT.

The RESTORE SYSTEM utility is run using LOGONLY SWITCH VCAT SYSVALUEDDN(ddname). This will bring the Db2 system to the required point in time.

Db2 Cloning Tool updates to the Db2 directory and catalog may have been regressed by the RESTORE SYSTEM, so the Db2 conditioning is done again.

Step overview

This section summarizes the steps you need to follow to perform Db2 subsystem cloning from the SLB to a specific point in time.

1. Run DB2GETBACKINFO to get the source data (set AD) and backup (set B1D) volume pairing used by the Db2 BACKUP SYSTEM command. (Any available generation can be specified; this example uses a token for the SLB of point in time 1.)
2. Run DB2GETBACKINFO to get the source log (set AL) and backup (set B2L) volume pairing used by the Db2 BACKUP SYSTEM command. (The generation specified here should be the one that follows the generation used in step 1; this example uses a token for the SLB of point in time 2.)
3. Run BACKINFO-REFORMAT to take the backinfo data sets created by the DB2GETBACKINFO commands (Steps 1 and 2) and reformat it for use by subsequent COPY commands (Steps 4 and 5).
4. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a USERCATALOGS-DDN keyword (data set from step 3), and include the VOLPLIST DD. If there are target volumes that have target ICF catalogs on them, then TARGET-UCATS-ON-TARGET-VOLUMES(Y) must be specified, and VOLPLIST DD should not be used. Otherwise, TARGET-UCATS-ON-TARGETVOLUMES should be set to N (or should not be specified), and VOLPLIST DD must be specified. If the VOLPLIST DD is specified, it must have attributes of RECFM=FB, LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.
5. Run COPY using FROM-USER-STORAGEGROUP and USERSGDEFS-DDN to get the backup volumes (set B) (data set from step 2), using the NOUSERCATALOGS keyword and including the VOLPLIST DD.
 - For the copy from volumes of the source storage group to volumes of the target storage group (for data or logs) using the same order of volumes, the KEEP-VOLUME-SEQUENCE(Y) parameter should be used.
 - If there are target volumes that have target ICF catalogs on them, then BASE-JOURNAL-DDN must be specified and must point to the journal from the first COPY command, and VOLPLIST DD should not be specified. Otherwise, specify VOLPLIST DD, and do not specify BASE-JOURNAL-DDN. If the VOLPLIST DD is specified, it must have attributes of RECFM=FB, LRECL=80.

The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.

6. If there are no target ICF catalogs on the target volumes, run CKZRNTGT with the VOLPLIST from step 4 on the CKZIN DD and the VOLPLIST from step 5 on the NUCIN DD. If there are target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) was used in step 4, omit this step.
7. If there are no target ICF catalogs on the target volumes, run VOLOPTIOINS using the NEWTGT data set as input to NEWTARGETS-DDN. If there are target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGETVOLUMES(Y) was used in step 4, omit this step.
8. Run RENAME for target volumes (set C).
9. Run DB2UPDATE to condition the BSDS and directory.
10. Create the conditional restart record for required point in time.
11. Start the Db2 systems.
12. Run the RESTORE SYSTEM utility using LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*).
13. Stop the Db2 systems.
14. Run the Db2 conditioning commands: DB2UPDATE, DB2ALTERBSDS(SLB-START), DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP, DB2UTILXCLEAN.

Note: Steps 4, 7, 8, 9, and 14 use the same journal data set. If there are target ICF catalogs on the target volumes, then step 5 uses two journal data sets: the first journal is the journal that was created in step 4 and the second journal is created in step 5. If there are no target ICF catalogs on the target volumes, then step 5 uses a different journal data set.

Example

In the example steps that follow, the following items are assumed:

- The source Db2 resides on the source volumes (SRCxxx). The source log volumes are in storage group SRCSTGL and the source data volumes are in storage group SRCSTGD.
- The target Db2 will reside on the target volumes (TGTxxx). The target log volumes are in storage group TGTSTGL and the target data volumes are in storage group TGTSTGD.
- There are at least two sets of backup volumes (BKPxxx) that have been created by Db2 BACKUP SYSTEM commands.
- There are two source ICF catalogs (USERCAT.SRCLG and USERCAT.SRCDB) that reside on source volumes. The USERCAT.SRCLG catalog is on a source log volume and contains entries for the data sets on the source log volumes. The USERCAT.SRCDB catalog is on a source data volume and contains entries for the data sets on the source data volumes.
- There are two target catalogs (USERCAT.TGTLG and USERCAT.TGTDB). It is possible one of the ICF catalogs resides on a target volume, both ICF catalogs reside on target volumes, or none of the target volumes contain ICF catalogs. The USERCAT.TGTLG catalog contains entries for the data sets on the target log volumes. The USERCAT.TGTDB catalog contains entries for the data sets on the target data volumes.
- There are two Db2 system level backups:
 - A backup for a point in time at 2013/07/06 11:10:15 that has a token of X'C5C2F1C1CB9E97E16641BD660000000000027217E83A'
 - A backup for a point in time at 2013/07/19 10:59:19 that has a token of X'C5C2F1C1CBAEEDAF121378A0000000000002ADD6A3F0'
- The intention is to create the target Db2 at a point in time of 2013/07/17 12:00:00.

Step 1

The first step is to get the source to backup data volume pairing (SRCxxx to BKPxxx) and identify the source ICF catalogs from the last Db2 BACKUP SYSTEM taken for location DB2PLOC for a point in time at 2013/07/06 11:10:15. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DSN=HLQ?.WRK.BACKINFO,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
DB2GETBACKINFO -
DATABASESONLY -
BACKINFO-DDN(BACKINFO) -
WORK-DDN(HSMLIST) -
TOKEN(X'C5C2F1C1CB9E97E16641BD660000000000027217E83A') -
LOCATION(DB2PLOC) -
USERCATALOGS( -
            USERCAT.SRCDB -
            )
//*
```

Step 2

This step retrieves the source to backup log volume pairing (SRCxxx to BKPxxx) and identifies the source ICF catalogs from the last Db2 BACKUP SYSTEM taken for location DB2PLOC for a point in time at

2013/07/19 10:59:19. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DSN=HLQ?.WRK.BACKINFL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
DB2GETBACKINFO          -
LOGSONLY                -
BACKINFO-DDN(BACKINFO) -
WORK-DDN(HSMLIST)       -
TOKEN('X'C5C2F1C1CBAEEDAF121378A0000000000002ADD6A3F0') -
LOCATION(DB2PLOC)        -
USERCATALOGS(          -
    USERCAT.SRCLG      -
)
//*
```

Step 3

This step reformats the outputs of steps 1 and 2 (backinfo data sets) for use in the COPY in steps 4 and 5. The user catalog pairs are also specified here. Sample JCL can be found in the installation library SCKZJCL in member CKZBKIRF.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFD
//          DD DISP=SHR,DSN=HLQ?.WRK.BACKINFL
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//USRSGDEF DD DSN=HLQ?.WRK.USRSGDEF,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS    DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
BACKINFO-REFORMAT          -
BACKINFO-DDN(BACKINFO)    -
VOLPAIRS-DDN(VOLPAIRS)    -
USRSGDEFS-DDN(USRSGDEF)   -
USERCATALOGS-DDN(UCATS)   -
USERCATALOGS(            -
    USERCAT.SRCLG USERCAT.TGTG -
    USERCAT.SRCDB USERCAT.TGTDB -
)
//*
```

Step 4

This step sets the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal, and backs up the source ICF catalogs from the backup volumes. Sample JCL can be found in the installation library SCKZJCL in member CKZCOPY.

Partial JCL for this step when there are target ICF catalogs on the target volumes follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD  DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD  DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECORG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//CKZIN   DD  *
COPY
  DATA-MOVER(PGM(NONE))          -
  VOLPAIRS-DDN(VOLPAIRS)         -
  USERCATALOGS-DDN(UCATS)        -
  SOURCESONLINE(N)               -
  TARGET-UCATS-ON-TARGET-VOLUMES(Y) -
  CATWORK-DSN(HLQ?.WRK.* )      -
  JOURNAL-DDN(JOURNAL)          -
//*
```

Partial JCL for this step when there are no target ICF catalogs on the target volumes follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD  DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD  DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECORG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN   DD  *
COPY
  DATA-MOVER(PGM(NONE))          -
  VOLPAIRS-DDN(VOLPAIRS)         -
  USERCATALOGS-DDN(UCATS)        -
  SOURCESONLINE(N)               -
  CATWORK-DSN(HLQ?.WRK.* )      -
  JOURNAL-DDN(JOURNAL)          -
//*
```

Step 5

This step copies the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs. Sample JCL can be found in the installation library SCKZJCL in member CKZCOPY.

Partial JCL for this step when there are target ICF catalogs on the target volumes follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD  DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECORG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//BASEJRNL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN   DD  *
COPY
  FROM-USER-STORAGEGROUP(
    SRCUGL SRCUGD
  )
  TO-STORAGEGROUP(
    TGTUGL TGTUGD
  )
  USRSGDEFS-DDN(USRSGDEF)
  USRSGDEFS-OFFSETS(VOLSER(1) SGNAM(8) INCL(18)) -
```

```

KEEP-VOLUMES-SEQUENCE(YES) -
NOUSERCATALOGS -
BASE-JOURNAL-DDN(BASEJRNL) -
JOURNAL-DDN(JOURNAL)
//*

```

Partial JCL for this step when there are no target ICF catalogs on the target volumes follows:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD DSN=HLQ?.NUCJRNL,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECOG=KS,KEYLEN=64,KEYOFF=0,
// LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECFM=FB,LRECL=80,BLKSIZE=0,
// SPACE=(CYL,(1,1))
//CKZIN DD *
COPY -
FROM-USER-STORAGEGROUP( -
SRCUGL SRCUGD -
) -
TO-STORAGEGROUP( -
TGUGL TGUGD -
) -
USRSGDEFS-DDN(USRSGDEF) -
USRSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLXCL(18)) -
KEEP-VOLUMES-SEQUENCE(YES) -
NOUSERCATALOGS -
JOURNAL-DDN(JOURNAL)
//*

```

Step 6

This step runs CKZRNTGT with the VOLPLIST data sets from steps 4 and 5. Sample JCL can be found in the installation library SCKZJCL in member CKZRNTGT.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 4), this step is omitted.

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A
//* USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE
//* NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIIONS COMMAND
//S2 EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
// DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
// SPACE=(CYL,(1,1))

```

Step 7

This step runs VOLOPTIIONS with the NEWTGT data set from step 6 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME. Sample JCL can be found in the installation library SCKZJCL in member CKZVOLOP.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 4), this step is omitted.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=6M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD  DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT  DD  DISP=SHR,DSN=HLQ?.WRK.NEWTGT
//CKZIN   DD  *
      VOLOPTIONS UPDATE      -
      NEWTARGETS-DDN(NEWTGT) -
      JOURNAL-DDN(JOURNAL)
//*
```

Step 8

This step RENAMEs the data sets on the target volumes (TGTxxx). Sample JCL can be found in the installation library SCKZJCL in member CKZREN.

Partial JCL and the RENAME command for this step follow:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG DD  SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD  SYSOUT=*
//JOURNAL DD  DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD  DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD  DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD  *
      RENAME                -
      SAFE                  -
      VOLBKUP-DDN(VOLBKUP)  -
      RENAME-MASKS(        -
          PROD1.** TEST1.** -
          PROD2.** TEST2.** -
          )                  -
      JOURNAL-DDN(JOURNAL)
//*
```

Step 9

This step conditions the BSDS and Db2 directory, and creates the SYSVALUE data for use by the later RESTORE SYSTEM step. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDUPD.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=6M
//STEPLIB DD  DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD  DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSVALUE DD  DSN=HLQ?.WRK.SYSVALUE,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//JOURNAL DD  DISP=SHR,DSN=HLQ?.JRNL
//BSDS01  DD  DISP=OLD,DSN=TEST1.BSDS01
//BSDS02  DD  DISP=OLD,DSN=TEST1.BSDS02
//SYSDBDXA DD  DISP=OLD,
//          DSN=TEST1.DSNDBC.DSNDB01.SYSDBDXA.I0001.A001
//CKZIN   DD  *
      DB2UPDATE              -
      DB2-HLQS(PROD1, TEST1, PROD2, TEST2 ) -
      DDF( . . . )           -
      HLQ-NOT-UPDATED(RC(22)) -
      SYSVALUE-DDN(SYSVALUE) -
//*
```

```
JOURNAL-DDN(JOURNAL)
//*
```

Step 10

This step creates the conditional restart record. This example is for a non-data sharing subsystem. The required point in time is 2013/07/17 12:00:00.

Partial JCL for this step follows:

```
//S1      EXEC PGM=DSNJU003,REGION=6M
//STEPLIB DD DISP=SHR,DSN=DSN.VB10.SDSNLOAD
//SYSUT1  DD DISP=SHR,DSN=TEST1.BSDS01
//SYSUT2  DD DISP=SHR,DSN=TEST1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
CRESTART CREATE,SYSPITRT=2013198120000
//*
```

Step 11

This step starts the Db2 system in special mode. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//        DD DISP=SHR,DSN=DB2.SDSNEXIT
//        DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN   DD *
DB2START
SSID(DB2T) -
REPLY-T0-RESTART-WTOR(Y) -
DSNZPARAM(specparm) -
SPECIAL
//*
```

Step 12

This step runs the RESTORE SYSTEM LOGONLY SWITCH VCAT. The SYSTEM RESTORE utility will apply log records to bring the target Db2 system to the desired point in time.

Partial JCL for this step follows:

```
//STEP1 EXEC DSNUPROC,TIME=1440,
// UTPROC=' ',
// SYSTEM='DB2T'
//SYSIN DD *
RESTORE SYSTEM LOGONLY SWITCH VCAT SYSVALUEDDN(SYSVALUE)
//*
//SYSVALUE DD DISP=SHR,DSN=HLQ?.WRK.SYSVALUE
//*
```

Step 13

This step stops the Db2 system. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTO.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//        DD DISP=SHR,DSN=DB2.SDSNEXIT
//        DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
```

```
//CKZIN DD *
DB2STOP -
SSID(DB2T)
//*
```

Step 14

This step runs DB2UPDATE DBD01ONLY to reapply updates to the Db2 directory that may have been regressed by the RESTORE SYSTEM. Sample JCL can be found in the installation library SCKZJCL in member CKZDUPD.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//SYSDBDXA DD DISP=OLD,
// DSN=TEST1.DSNDBC.DSNDB01.SYSDBDXA.I0001.A001
//CKZIN DD *
DB2UPDATE -
DBD01ONLY -
DB2-HLQS(PROD1,TEST1,PROD2,TEST2) -
DDF( . . . ) -
HLQ-NOT-UPDATED(RC(22)) -
JOURNAL-DDN(JOURNAL)
//*
```

Step 15

This step starts the Db2 system in special mode. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=DB2.SDSNEXIT
// DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2START -
SSID(DB2T) -
DSNZPARAM(specparm) -
SPECIAL
//*
```

Step 16

This step runs DB2FIX DATABASES(DB2) to resolve any LPL or GRECP status on the Db2 directory and catalog. Sample JCL can be found in the installation library SCKZJCL in member CKZDFIX.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=DB2.SDSNEXIT
// DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2FIX -
SSID(DB2T) -
DATABASES(DB2)
//*
```

If DB2FIX indicates that page space SYSDBDXA in database DSNDB01 was started due to having LPL or GRECP status, as indicated by the return code and by the message CKZ23526E DSNDB01.SYSDBDXA IS

IN RESTRICTED STATUS; DB2UPDATE NEEDS TO BE RUN AGAIN, the changes made to SYSDBDXA by DB2UPDATE may have been regressed and need to be redone. The target Db2 subsystem must be stopped, DB2UPDATE run again using the DBD01ONLY keyword, and the Db2 subsystem started again in maintenance mode using the special zparms.

Step 17

This step runs DB2SQL to update the Db2 catalog. Sample JCL can be found in the installation library SCKZJCL in member CKZDSQL.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN   DD *
    DB2SQL                -
    SSID(DB2T)             -
    WLM-ENVIRONMENT-MASKS(PROD* TEST*) -
    JOURNAL-DDN(JOURNAL)
//*
```

Step 18

This step runs DB2FIX DATABASES(APPLICATION) to resolve any LPL or GRECP status on the application table and index spaces. Sample JCL can be found in the installation library SCKZJCL in member CKZDFIX.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//        DD DISP=SHR,DSN=DB2.SDSNEXIT
//        DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN   DD *
    DB2FIX                -
    SSID(DB2T)             -
    DATABASES(APPLICATION)
//*
```

Step 19

This step generates and runs the SQL statements that are necessary to re-create the objects that cannot be processed by the Db2 CATMAINT utility with new schema values (views, materialized query tables, SQL scalar functions, triggers, and native SQL procedures), and invokes the Db2 CATMAINT utility to change schema values for the rest of the objects. Sample JCL can be found in the installation library SCKZJCL in member CKZDUPDS.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//        DD DSN=DSNxxx.SDSNLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DSN=CKZ.JRNL,DISP=OLD
//DRDDL   DD DSN=CKZ.DRDDL
//CRDDL   DD DSN=CKZ.CRDDL
//CKZIN   DD *
    DB2SCHEMA-UPDATE      -
    DB2-SSID(DB2T)        -
    DROP-DDL-DDN(DRDDL)   -
    CREATE-DDL-DDN(CRDDL) -
    SCHEMA-MASKS(SRCSCHM TRGSCHM) -
    JOURNAL-DDN(JOURNAL)
```

Step 20

This step stops the Db2 system. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTO.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=DB2.SDSNEXIT
// DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2STOP -
SSID(DB2T)
//*
```

Step 21

This step is optional. This step is only used for the cloning of a Db2 system when it is desired to remove utility information from the target Db2 subsystem. This step will remove all entries from SYSUTILX. Sample JCL can be found in the installation library SCKZJCL in member CKZDUTCL.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=DB2.SDSNEXIT
// DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN DD *
DB2UTILXCLEAN -
JOURNAL-DDN(JOURNAL)
//*
```

Step 22

This step starts the Db2 system in normal mode. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=DB2.SDSNEXIT
// DD DISP=SHR,DSN=DB2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2START -
SSID(DB2T) -
NORMAL
//*
```

Considerations for data sharing

To perform these cloning steps to a specific point in time from a Db2 BACKUP SYSTEM backup in a data sharing environment, note the following:

- Step 9 runs DB2UPDATE for all members. The DB2UPDATE for the secondary members should include BSDSONLY.
- Step 10 creates conditional restart records in each member.
- Step 11 starts all members in MAINT mode. The DB2START commands should include the keyword STOP-WAITING-IF-DSNR02OI(Y,RC(3)).

- Step 13 stops all members.
- In step 15, only one member must be started.
- Step 22 starts all members in normal mode.

Db2 subsystem cloning from Db2 BACKUP SYSTEM dump tapes

Db2 subsystems can be cloned by using the backup dump tapes created by a Db2 BACKUP SYSTEM command.

Overview

Because the source Db2 is running at the time of the Db2 BACKUP SYSTEM command, this is an online cloning. This situation is similar to cloning using an interim set of volumes. Db2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes.

If the target volumes are set up in storage groups in a way that mirrors the source volume configuration, it is possible to pair the backup volumes to the target volumes by using the source and target storage group names. For example, if the source has two storage groups, one for logs (SRCSGL) and one for data (SRCSGD), and the target has two storage groups, one for logs (TGTSGL) and one for data (TGTSGD), then the dump tapes can be paired to the target volumes by using the source and target storage group names. The dump tapes that correspond to the source log storage group (SRCSGL) are paired with volumes in the target log storage group (TGTSGL), and the dump tapes that correspond to the source data storage group (SRCSGD) are paired with volumes in the target data storage group (TGTSGD).

The following procedure allows for repetitive cloning without requiring a manual update of the volume specifications when the LAST keyword is used with the DB2GETBACKINFO command.

- The source volumes (referred to as set A) have been copied to the backup dump tapes (set T) by a Db2 BACKUP SYSTEM command.
- The dump tapes (set T) are restored to the target volumes (set C), re-labeled with the target volume volser, and brought back online.
- The data sets on the target volumes (set C) are renamed.
- The target Db2 subsystems on the target volumes are conditioned.
- Db2 Cloning Tool requires the original source volumes (set A) and the final target volumes (set C) to rename and catalog the target volume data sets and to update the target Db2 system to replace the original source volumes with the final target volumes. To accomplish this, Db2 Cloning Tool must know the pairing between the source volumes (set A) and the target volumes (set C).
- To obtain the original source volumes (set A) and the backup dump tapes (set T) from DFSMSHsm, the DB2GETBACKINFO command is executed to query DFSMSHsm. Any generation can be selected in the DB2GETBACKINFO command. In this example, LAST is used for the most current version pairs of set A and set T. This list is provided in DFSMSHsm LIST format in HSMLIST DD. Db2 Cloning Tool also creates the BACKINFO DD data set in a format that is consistent with a utility in Db2 Recovery Expert and Rocket Mainstar Database Backup and Recovery for Db2 on z/OS (DBR for Db2), only for the purpose of keeping cloning from a system backup consistent across products. The HSMLIST DD is not used again. The source ICF catalogs must also be identified by the user in this step.
- The job that executes the RESTORE-FROM-DUMPTAPES command does several things:
 - Reads the BACKINFO DD data set created in the previous step to:
 - Pair target volumes using TO-VOLSER or TO-STORAGEGROUP, along with volume capacity and the order in which the volumes (or storage groups) are specified. If KEEP-VOLUMES-SEQUENCE(YES) is specified, the SOURCE-STORAGEGROUP parameter must be specified and include the source storage groups, and the TO-STORAGEGROUP parameter must be specified and include the target storage groups. In addition, the position number of the source volume that is defined in the SOURCE-STORAGEGROUP must be equal to the position number of the target volume defined in the TO-STORAGEGROUP. For example, to map volumes from SRCGRP1 to target storage group TGTGRP1, you must enter the storage group names in the same order in both of these keywords: SOURCE-STORAGEGROUP(SRCGRP1, SRCGRP2) and TO-STORAGEGROUP(TGTGRP1, TGTGRP2).

- If there are target ICF catalogs on the target volumes and TARGET-UCATS-ON-TARGET-VOLUMES(Y) is specified, then CAS is used to issue UNALLOCATE commands for the target ICF catalogs that reside on target volumes, and only the source volumes that have catalogs on them are paired with the target volumes that have catalogs on them.
- Build and run ADRDSSU RESTORE commands to restore all eligible Source volumes.
- Selects target volumes using TO-VOLSER and/or TO-STORAGEGROUP, along with volume capacity and the order in which the volumes (or storage groups) are specified to match volumes.
- Creates the VOLPAIRS DD data set with volume pairs as determined above. These files will be used in the subsequent COPY step.
- Creates the STATUS file to provide resume capability.
- The user catalogs are added to this step and stored in the data set pointed to by UCATS DDN so that they can be passed to the COPY command.

For this procedure, one COPY command is used. The COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword, which does not copy any volumes; the target volumes are created by the RESTORE-FROM-DUMPTAPES command, which restores the backup dump tapes onto the target volumes.
- Identifies the source volume (set A) to target volume (set C) pairing by reading the VOLPAIRS-DDN created in the previous step.
- Backs up the source ICF catalog copies from the target volumes.
- Creates a journal data set, which is used throughout the RENAME and Db2 conditioning steps.

The RENAME renames and catalogs all of the target volume data sets (set C) to new names.

The Db2 conditioning updates the appropriate Db2 data sets on the target volumes (set C) to the new data set names and volume serials.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the Db2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

For data sharing, the first members started will receive a DSNR020I WTOR due to the conditional restart records contained in the other members. If the start jobs are being run serially by a job scheduler, it is recommended that the DB2START command for all but the last member include the STOP-WAITING-IF-DSNR020I keyword with a value of Y. This will allow the starts of the other members to be done concurrently. After the last member has been started, a DB2START for the first member should be done again using the WAITONLY and DB2-ALREADY-RUNNING(RC(0)) keywords. Completion of the DB2START with a return code of 4 or less indicates that the first member is up and ready for the remainder of the Db2 conditioning jobs.

Step overview

This section summarizes the steps you need to follow to perform Db2 subsystem cloning using the Db2 BACKUP SYSTEM dump tapes.

1. Run DB2GETBACKINFO to get the source (set A) and backup dump tape (set T) volume pairing used by the Db2 BACKUP SYSTEM command. (Any available generation can be specified; our example uses LAST.)
2. Run RESTORE-FROM-DUMPTAPES to take the backinfo data set created by the DB2GETBACKINFO command (Step 1) and generate and run ADRDSSU restore commands, and create metadata for use by the subsequent COPY command (Step 3). KEEP-VOLUMES-SEQUENCE(YES) should be specified for pairing of volumes by source and target storage group sequence numbers from the SOURCE-STORAGEGROUP and TO-STORAGEGROUP parameters. If target ICF catalogs reside on target volumes, then TARGET-UCATS-ON-TARGET-VOLUMES(Y) should be specified.

3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to target volumes (set C) pairing (data set from step 2), and use a USERCATALOGS-DDN keyword (data set from step 2). If target ICF catalogs reside on target volumes, then TARGET-UCATS-ON-TARGET-VOLUMES(Y) should be specified.
4. Run RENAME for target volumes (set C).
5. Run the Db2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP.

Example

In the example steps that follow, the following items are used:

- The source Db2 resides on the source volumes (SRCxxx). The source log volumes are in storage group SRCSGL and the source data volumes are in storage group SRCSGD.
- The target Db2 will reside on the target volumes (TGTxxx). The target log volumes are in storage group TGTSGL and the target data volumes are in storage group TGTSGD.
- There are one or more sets of backup dump tapes (DMPxxx) that have been created by Db2 BACKUP SYSTEM commands.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.

Step 1

The first step is to get the source volume to backup dump tape pairing (SRCxxx to DMPxxx) and identify the source ICF catalogs from the last Db2 BACKUP SYSTEM taken for location DB2PLOC. Sample JCL can be found in the installation SCKZJCL library in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD   DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD   DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//BACKINFO DD  DSN=HLQ?.WRK.BACKINFO,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD   DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD   *
DB2GETBACKINFO          -
  BACKINFO-DDN(BACKINFO) -
  WORK-DDN(HSMLIST)      -
  LAST                   -
  LOCATION(DB2PLOC)     -
  USE-DUMPTAPES         -
  USERCATALOGS(        -
    USERCAT.SRC01      -
    USERCAT.SRC02      -
  )
//*
```

Step 2

The second step is to restore the dump tapes of the source volumes to the target volumes or storage groups, using the output of Step 1 (BACKINFO data set). The user catalog pairs are also specified in this step. Sample JCL can be found in the installation SCKZJCL library in member CKZRSTDT.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD   DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD   DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD  SYSOUT=*
```

```

//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DD DISP=(,CATLG),UNIT=SYSALLDA,
//          DD SPACE=(CYL,(1,1))
//STATUS   DD DSN=HLQ?.WRK.STATUS,
//          DD DISP=(,CATLG),SPACE=(CYL,(1,1)),
//          DD RECOG=KS,KEYLEN=64,KEYOFF=0,
//          DD LRECL=600,UNIT=SYSALLDA
//UCATS    DD DSN=HLQ?.WRK.UCATS,
//          DD DISP=(,CATLG),UNIT=SYSALLDA,
//          DD SPACE=(CYL,(1,1))
//CKZIN    DD *
//          RESTORE-FROM-DUMPTAPES -
//          SOURCE-STORAGEGROUP( SRCUGL SRCUGD ) -
//          TO-STORAGEGROUP( TGTUGL TGTUGD ) -
//          BACKINFO-DDN(BACKINFO) -
//          STATUS-DDN(STATUS) -
//          VOLPAIRS-DDN(VOLPAIRS) -
//          USERCATALOGS-DDN(UCATS) -
//          USERCATALOGS( -
//              USERCAT.SRC01 USERCAT.TGT01 -
//              USERCAT.SRC02 USERCAT.TGT02 -
//          )
//*

```

Step 3

The third step is to set the pairing between source (SRCxxx) and target (TGTxxx) volumes in the journal and back up the source ICF catalogs from the target volumes. Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1       EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI   DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS    DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL  DD DSN=HLQ?.JRNL,
//          DD DISP=(,CATLG),UNIT=SYSALLDA,
//          DD RECOG=KS,KEYLEN=64,KEYOFF=0,
//          DD LRECL=600,SPACE=(CYL,(10,10))
//CKZIN    DD *
//          COPY -
//          DATA-MOVER(PGM(NONE)) -
//          VOLPAIRS-DDN(VOLPAIRS) -
//          USERCATALOGS-DDN(UCATS) -
//          CATWORK-DSN(HLQ?.WRK.* ) -
//          JOURNAL-DDN(JOURNAL)
//*

```

Step 4

The fourth step is the RENAME of the data sets on the target volumes (TGTxxx). Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```

//S1       EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI   DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG  DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS  DD SYSOUT=*
//JOURNAL  DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS  DD DSN=HLQ?.WRK.BCSRECS,
//          DD UNIT=SYSALLDA,DISP=(,CATLG),
//          DD SPACE=(CYL,(10,10))
//VOLBKUP  DD DSN=HLQ?.WRK.VOLBKUP,
//          DD UNIT=SYSALLDA,DISP=(,CATLG),
//          DD SPACE=(CYL,(40,40))
//CKZIN    DD *
//          RENAME -

```

```

SAFE -
VOLBKUP-DDN(VOLBKUP) -
RENAME-MASKS( -
    PRODLG.** TESTLG.** -
    PROddb.** TESTDB.** -
) -
JOURNAL-DDN(JOURNAL)
/**

```

Step 5

The last step is to run the Db2 conditioning commands. These commands are:

- DB2UPDATE
- DB2ALTERBSDS SLB-START
- DB2START
- DB2FIX DATABASES(DB2)
- DB2SQL
- DB2FIX DATABASES(APPLICATION)
- DB2SCHEMA-UPDATE
- DB2STOP

For additional information about the conditioning commands, refer to the appropriate section in [“Db2 online cloning”](#) on page 134 for your specific type of cloning.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the Db2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

The following are examples of the DB2UPDATE, DB2ALTERBSDS, and DB2START steps necessary to handle this condition.

Example 1 – a non-data sharing subsystem with active logs that are striped:

1. Update the Db2 directory and the BSDS.

```

DB2UPDATE -
DB2-HLQS( srcvcat tgtvcat ) -
DDF( ... ) -
JOURNAL-DDN(JOURNAL)

```

2. Create a system-level backup (SLB) start conditional restart record in the BSDS.

```

DB2ALTERBSDS -
SLB-START -
JOURNAL-DDN(JOURNAL)

```

3. Start the target Db2 in special mode and automatically reply to the Db2 restart WTOR.

```

DB2START -
DB2-SSID(ssid) -
SPECIAL -
DSNZPARM(ssidSPEC) -
REPLY-TO-RESTART-WTOR(Y)

```

Example 2 - a data sharing group with three members (the active logs may or may not be striped):

1. Update the Db2 directory and the BSDS for mbr1.

```

DB2UPDATE -
DB2-HLQS( srcvcat tgtvcat ) -
DB2-GROUP( srcgrp tgtgrp ) -
DB2-MEMBERS( srcmbr1 mbr1 -

```

```

srcmbr2 mbr2 -
srcmbr3 mbr3 ) -
DDF( ... ) -
JOURNAL-DDN(JOURNAL)

```

2. Update the BSDS for mbr2.

```

DB2UPDATE -
BSDSONLY -
DB2-HLQS( srcvcat tgtvcat ) -
DB2-GROUP( srcgrp tgtgrp ) -
DB2-MEMBERS( srcmbr1 mbr1 -
srcmbr2 mbr2 -
srcmbr3 mbr3 ) -
DDF( ... ) -
JOURNAL-DDN(JOURNAL)

```

3. Update the BSDS for mbr3.

```

DB2UPDATE -
BSDSONLY -
DB2-HLQS( srcvcat tgtvcat ) -
DB2-GROUP( srcgrp tgtgrp ) -
DB2-MEMBERS( srcmbr1 mbr1 -
srcmbr2 mbr2 -
srcmbr3 mbr3 ) -
DDF( ... ) -
JOURNAL-DDN(JOURNAL)

```

4. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr1.

```

DB2ALTERBSDS -
DB2-MEMBER(mbr1) -
SLB-START -
JOURNAL-DDN(JOURNAL)

```

5. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr2.

```

DB2ALTERBSDS -
DB2-MEMBER(mbr2) -
SLB-START -
JOURNAL-DDN(JOURNAL)

```

6. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr3.

```

DB2ALTERBSDS -
DB2-MEMBER(mbr3) -
SLB-START -
JOURNAL-DDN(JOURNAL)

```

7. Start the target Db2 mbr1 in special mode, automatically reply to the Db2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

DB2START -
DB2-SSID(mbr1) -
SPECIAL -
DSNZPARM(mbr1SPEC) -
REPLY-TO-RESTART-WTOR(Y) -
STOP-WAITING-IF-DSNR020I(Y)

```

8. Start the target Db2 mbr2 in special mode, automatically reply to the Db2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

DB2START -
DB2-SSID(mbr2) -
SPECIAL -
DSNZPARM(mbr2SPEC) -
REPLY-TO-RESTART-WTOR(Y) -
STOP-WAITING-IF-DSNR020I(Y,RC(3))

```

9. Start the target Db2 mbr3 in special mode and automatically reply to the Db2 restart WTOR.

```

DB2START -
DB2-SSID(mbr3) -

```

```
SPECIAL -
DSNZPARM(mbr3SPEC) -
REPLY-TO-RESTART-WTOR(Y)
```

10. Stop the target Db2 mbr3.

```
DB2STOP -
DB2-SSID(mbr3)
```

11. Stop the target Db2 mbr2.

```
DB2STOP -
DB2-SSID(mbr2)
```

12. Wait for the target Db2 mbr1 to complete its startup.

```
DB2START -
DB2-SSID(mbr1) -
SPECIAL -
DSNZPARM(mbr1SPEC) -
REPLY-TO-RESTART-WTOR(Y) -
DB2-ALREADY-RUNNING(RC(0)) -
WAITONLY
```

Db2 subsystem cloning from other system level backups when backup volumes are online

Db2 subsystems can be cloned using the backup volumes created by one of the following backup products: Db2 Recovery Expert for z/OS, Rocket System Backup and Recovery for Db2 (RBR), or Rocket Mainstar Database Backup and Recovery for Db2 on z/OS (DBR for Db2). Because the source Db2 is running at the time of the backup, this is an online cloning.

For the purposes of this procedure, the following acronyms are used and should be substituted where the variable *ccc* is used:

- Db2 Recovery Expert: *ccc*=ARY
- RBR: *ccc*=RBR
- DBR for Db2: *ccc*=DBR

The following procedure allows for repetitive cloning without the need to manually update the volume specifications when LAST is used with the *ccc*#VOLS program. The *ccc*#VOLS program is part of the Db2 Recovery Expert, RBR, and DBR for Db2 products.

- The source volumes (referred to as set A) have been copied to the backup volumes (set B) by a Db2 Recovery Expert, RBR, or DBR for Db2 backup.
- The backup volumes (set B) are copied to the target volumes (set C).
- The data sets on the target volumes (set C) are renamed.
- The target Db2 subsystems on the target volumes are conditioned.

For this procedure, the source to backup volume pairing will be obtained from Db2 Recovery Expert, RBR, or DBR for Db2 and reformatted for use by two COPY commands. Each COPY command must be in its own JCL step.

- The first COPY command identifies the source volume to backup volume pairing and backs up the source ICF catalog copies on the backup volumes. This COPY does not do any volume copies as the copies were done by the Db2 Recovery Expert, RBR, or DBR for Db2 backup.
- The second COPY command copies the backup volumes to the target volumes and does not back up the ICF catalogs.

If target ICF catalogs reside on the target volumes (set C) and source ICF catalogs reside on the source volumes (set A), then the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option should be used in the first COPY command, and the BASE-JOURNAL-DDN from the first COPY command should be used in the second COPY command.

If there are no target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) is not used for the first COPY command, then an additional DD must be added to the COPY steps. The additional DD name is VOLPLIST and it must have attributes of RECFM=FB, LRECL=80. The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

1. Run the ccc#VOLS program to get the source (set A) and backup (set B) volume pairing used by the Db2 Recovery Expert, RBR, or DBR for Db2 backup and the names of the source ICF user catalogs.
2. Run BACKINFO-REFORMAT to take the backinfo data set created by the ccc#VOLS program (Step 1) and reformat it for use by subsequent COPY commands (Steps 3 and 4).
3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a USERCATALOGS-DDN keyword (data set from step 2), and include the VOLPLIST DD. If there are target volumes that have target ICF catalogs on them, then TARGET-UCATS-ON-TARGET-VOLUMES(Y) must be specified, and VOLPLIST DD should not be used. Otherwise, TARGET-UCATS-ON-TARGET-VOLUMES should be set to N (or should not be specified) and VOLPLIST DD must be specified. If the VOLPLIST is specified, it must have attributes of RECFM=FB, LRECL=80. The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.
4. Run COPY using FROM-USER-STORAGEGROUP and USERSGDEFS-DDN to get the backup volumes (set B) (data set from step 2), using the NOUSERCATALOGS keyword and including the VOLPLIST DD.
 - For the copy from volumes of the source storage group to volumes of the target storage group (for data or logs) using the same order of volumes, the KEEP-VOLUMES-SEQUENCE(Y) parameter should be used.
 - If there are target volumes that have target ICF catalogs on them, then BASE-JOURNAL-DDN must be specified and must point to the journal from the first COPY command, and VOLPLIST DD should not be specified. Otherwise, specify the VOLPLIST DD, and do not specify BASE-JOURNAL-DDN. If the VOLPLIST is specified it must have attributes of RECFM=FB,LRECL=80.

The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.

5. If there are no target ICF catalogs on the target volumes, run CKZRNTGT with the VOLPLIST from step 3 on the CKZIN DD and the VOLPLIST from step 4 on the NUCIN DD. If there are target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) was used in step 3, omit this step.
6. If there are no target ICF catalogs on the target volumes, run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN. If there are target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGETVOLUMES(Y) was used in step 3, omit this step.
7. Run RENAME for target volumes (set C).
8. Run the Db2 conditioning commands: DB2UPDATE, DB2ALTERBSDS(SLB-START), DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP, DB2UTILXCLEAN.

Note: Steps 3, 6, 7, and 8 use the same journal data set. If there are target ICF catalogs on the target volumes, then step 4 uses two journal data sets: the first journal is the journal that was created in step 3 and the second journal is created in step 4. If there are no target ICF catalogs on the target volumes, then step 4 uses a different journal data set.

Example

In this example:

- The source Db2 resides on the source volumes (SRCxxx)
- The target Db2 will reside on the target volumes (TGTxxx).
- There are one or more sets of backup volumes (BKPxxx) that have been created by Db2 Recovery Expert, RBR, or DBR for Db2 backups.

- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.
- There are two target ICF catalogs (USERCAT.TGT01 and USERCAT.TGT02). It is possible that one of the ICF catalogs resides on a target volume, both ICF catalogs reside on target volumes, or none of the target volumes contain ICF catalogs.

Step 1 - Get the source to backup volume pairing (SRCxxx to BKPxxx) and the source ICF catalog names from the last backup taken for Db2 subsystem DB2P.

Run the ccc#VOLS program to get the source (set A) and backup (set B) volume pairing used by the Db2 Recovery Expert, RBR, or DBR for Db2 backup and the names of the source ICF user catalogs. Sample JCL for this program can be found in the Db2 Recovery Expert, RBR, or DBR for Db2 ScccSAMP library.

Step 2 - Reformat the output of step 1 (backinfo data set) for use in the COPY in steps 3 and 4.

The user catalog pairs are also specified here. Sample JCL can be found in the installation SCKZJCL library in member CKZBKIRF.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//FRVOLSER DD DSN=HLQ?.WRK.FRVOLSER,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
BACKINFO-REFORMAT          -
BACKINFO-DDN(BACKINFO)    -
VOLPAIRS-DDN(VOLPAIRS)    -
FROM-VOLSER-DDN(FRVOLSER) -
USERCATALOGS-DDN(UCATS)   -
USERCATALOGS(             -
  USERCAT.SRC01 USERCAT.TGT01 -
  USERCAT.SRC02 USERCAT.TGT02 -
)
//*
```

Step 3 - Set the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal and back up the source ICF catalogs from the backup volumes.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step when there are target ICF catalogs on the target volumes follows:

```
//S1      EXEC PGM=000010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//CKZIN   DD *
COPY
  DATA-MOVER(PGM(NONE)) -
  VOLPAIRS-DDN(VOLPAIRS) -
```

```

USERCATALOGS-DDN(UCATS) -
TARGET-UCATS-ON-TARGET-VOLUMES(Y) -
SOURCESONLINE(N) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*

```

Partial JCL for this step when there are no target ICF catalogs on the target volumes follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECOG=KS,KEYLEN=64,KEYOFF=0,
//         LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECFM=FB,LRECL=80,BLKSIZE=0,
//         SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
DATA-MOVER(PGM(NONE)) -
VOLPAIRS-DDN(VOLPAIRS) -
USERCATALOGS-DDN(UCATS) -
SOURCESONLINE(N) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*

```

Step 4 - Copy the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step when there are target ICF catalogs on the target volumes follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECOG=KS,KEYLEN=64,KEYOFF=0,
//         LRECL=600,SPACE=(CYL,(10,10))
//BASEJRNL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN   DD *
COPY
FROM-USER-STORAGEGROUP(
)
USRSGDEFS-DDN(USRSGDEF)
USRSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLEXCL(18))
TO-STORAGEGROUP(
)
KEEP-VOLUMES-SEQUENCE(YES)
NOUSERCATALOGS
BASE-JOURNAL-DDN(BASEJRNL)
JOURNAL-DDN(JOURNAL)
//*

```

Partial JCL for this step when there are no target ICF catalogs on the target volumes follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//FRVOLSER DD DISP=SHR,DSN=HLQ?.WRK.FRVOLSER
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECOG=KS,KEYLEN=64,KEYOFF=0,

```

```

//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN    DD *
COPY                                -
  FROM-USER-STORAGEGROUP(          -
                                SRCUGL SRCUGD -
                                )        -
  USERSGDEFS-DDN(USRSGDEF)        -
  USERSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLEXCL(18)) -
  TO-STORAGEGROUP(                -
                                TGTUGL TGTUGD -
                                )        -
  KEEP-VOLUMES-SEQUENCE(YES)      -
  NOUSERCATALOGS                  -
  JOURNAL-DDN(JOURNAL)            -
//*

```

Step 5 - Run CKZRNTGT with the VOLPLIST data sets from steps 3 and 4.

Sample JCL can be found in the installation sCKZJCL library in member CKZRNTGT.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 3), this step is omitted.

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A USERCATALOGS KEYWORD,
//* DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE NOUSERCATALOGS KEYWORD,
//* DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIIONS COMMAND
//S2 EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//          DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//

```

Step 6 - Run VOLOPTIIONS with the newtgt data set from step 5 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME.

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 3), this step is omitted.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=SHR
//CKZIN DD *
  VOLOPTIIONS UPDATE              -
  NEWTARGETS-DDN(NEWTGT)         -
  JOURNAL-DDN(JOURNAL)           -
//*

```

Step 7 - RENAME the data sets on the target volumes (TGTxxx).

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD *
  RENAME          -
  SAFE            -
  VOLBKUP-DDN(VOLBKUP) -
  RENAME-MASKS(   -
    PROD1.** TEST1.** -
    PROD2.** TEST2.** -
  )              -
  JOURNAL-DDN(JOURNAL)
//*
```

Step 8 - Run the Db2 conditioning commands.

Run the Db2 conditioning commands: DB2UPDATE, DB2ALTERBSDS(SLB-START), DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP, DB2UTILXCLEAN. For additional information about the conditioning commands, refer to the [“Db2 online cloning”](#) on page 134 section for your specific type of cloning.

Db2 subsystem cloning from other system level backups when backup volumes are offline

Db2 subsystems can be cloned using the backup volumes created by one of the following backup products: Db2 Recovery Expert for z/OS, Rocket System Backup and Recovery for Db2 (RBR), or Rocket Mainstar Database Backup and Recovery for Db2 on z/OS (DBR for Db2). The offline backup volumes are exact copies of the source volumes and have an internal volsr that is the same as a source volsr. Because the source Db2 is running at the time of the backup, this is an online cloning.

For the purposes of this procedure, the following acronyms are used and should be substituted where the variable *ccc* is used:

- Db2 Recovery Expert: *ccc*=ARY
- RBR: *ccc*=RBR
- DBR for Db2: *ccc*=DBR

The following procedure allows for repetitive cloning without the need to manually update the volume specifications when LAST is used with the *ccc*#VOLS program. The *ccc*#VOLS program is part of the Db2 Recovery Expert, RBR, and DBR for Db2 products.

- The source volumes (referred to as set A) have been copied to the backup volumes (set B) by a Db2 Recovery Expert, RBR, or DBR for Db2 backup.
- The backup volumes (set B) are clipped with unique volsers and varied online.
- The backup volumes (set B) are copied to the target volumes (set C).
- The backup volumes (set B) are varied offline and clipped back to their original source volsr.
- The data sets on the target volumes (set C) are renamed.
- The target Db2 subsystems on the target volumes are conditioned.

For this procedure, the source to backup volume pairing will be obtained from Db2 Recovery Expert, RBR, or DBR for Db2 and reformatted for use by two COPY commands. Each COPY command must be in its own JCL step.

- The first COPY command identifies the source volume to backup volume pairing and backs up the source ICF catalog copies on the backup volumes. This COPY does not do any volume copies as the copies were done by the Db2 Recovery Expert, RBR, or DBR for Db2 backup.
- The second COPY command copies the backup volumes to the target volumes and does not back up the ICF catalogs.

If target ICF catalogs reside on the target volumes (set C) and source ICF catalogs reside on the source volumes (set A), then the TARGET-UCATS-ON-TARGET-VOLUMES(Y) option should be used in the first COPY command, and the BASE-JOURNAL-DDN from the first COPY command should be used in the second COPY command.

If there are no target ICF catalogs on the target volumes and the TARGET-UCATS-ON-TARGET-VOLUMES(Y) is not used for the first COPY command, then an additional DD must be added to the COPY steps. The additional DD name is VOLPLIST and it must have attributes of RECFM=FB, LRECL=80. The data set that is allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

1. Run the `ccc#VOLS` program to get the source (set A) and backup (set B) volume pairing used by the Db2 Recovery Expert, RBR, or DBR for Db2 backup and the names of the source ICF user catalogs.
2. Run `BACKINFO-REFORMAT` to take the `backinfo` data set created by the `ccc#VOLS` program (Step 1) and reformat it for use by subsequent COPY commands (Steps 3 and 4). The `CLIP-IF-OFFLINE(Y)` is used to request that the offline backup volumes (set B) be clipped and varied online. The `VOLSER-RENAME-MASKS` keyword is used to specify how the source volsers should be changed to generate unique volsers for the corresponding backup volumes being clipped. The `VOLOPTIONS-CMD-DDN` keyword is specified to generate a `VOLOPTIONS` command to unclip the backup volumes after they have been copied to the target volumes.
3. Run COPY with `DATA-MOVER(PGM(NONE))`. Use `VOLPAIRS-DDN` to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a `USERCATALOGS-DDN` keyword (data set from step 2), and include the `VOLPLIST DD`. If there are target volumes that have target ICF catalogs on them, then `TARGET-UCATS-ON-TARGET-VOLUMES(Y)` must be specified, and `VOLPLIST DD` should not be used. Otherwise, `TARGET-UCATS-ON-TARGET-VOLUMES` should be set to N (or should not be specified) and `VOLPLIST DD` must be specified. If the `VOLPLIST` is specified, it must have attributes of `RECFM=FB, LRECL=80`. The data set that is allocated by the `VOLPLIST DD` will be used by a subsequent step/job.
4. Run COPY using `FROM-VOLSER-DDN` to get the backup volumes (set B) (data set from step 2), using the `NOUSERCATALOGS` keyword and including the `VOLPLIST DD`. If there are target volumes that have target ICF catalogs on them, then `BASE-JOURNAL-DDN` must be specified and must point to the journal from the first COPY command, and `VOLPLIST DD` should not be specified. Otherwise, specify `VOLPLIST DD` and do not specify `BASE-JOURNAL-DDN`. If the `VOLPLIST DD` is specified, it must have attributes of `RECFM=FB, LRECL=80`. The data set that is allocated by the `VOLPLIST DD` will be used by a subsequent step/job.
5. Run `VOLOPTIONS` with `CKZIN` using the data set created in step 2 by the `VOLOPTIONS-CMD-DDN` keyword to vary offline and unclip the backup volumes (set B). The journal data set created in step 3 is used.
6. If there are no target ICF catalogs on the target volumes, run `CKZRNTGT` with the `VOLPLIST` from step 3 on the `CKZIN DD` and the `VOLPLIST` from step 4 on the `NUCIN DD`. If there are target ICF catalogs on the target volumes and the `TARGET-UCATS-ON-TARGET-VOLUMES(Y)` was used in step 3, omit this step.
7. If there are no target ICF catalogs on the target volumes, run `VOLOPTIONS` using the `NEWTGT` data set from Step 6 as input to `NEWTARGETS-DDN`. If there are target ICF catalogs on the target volumes and the `TARGET-UCATS-ON-TARGETVOLUMES(Y)` was used in step 3, omit this step.

8. Run CKZRNTGT with the VOLPLIST from step 3 on the CKZIN DD and the VOLPLIST from step 4 on the NUCIN DD.
9. Run VOLOPTIONS using the NEWTGT data set from step 6 as input to NEWTARGETS-DDN.
10. Run RENAME for target volumes (set C).
11. Run the Db2 conditioning commands: DB2UPDATE, DB2ALTERBSDS(SLB-START), DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP, DB2UTILXCLEAN.

Note: Steps 3, 5, 7, and 9 use the same journal data set. If there are target ICF catalogs on the target volumes, then step 4 uses two journal data sets: the first journal is the journal that was created in step 3 and the second journal is created in step 4. If there are no target ICF catalogs on the target volumes, then step 4 uses a different journal data set.

Example

In this example:

- The source Db2 resides on the source volumes (SRCxxx)
- The target Db2 will reside on the target volumes (TGTxxx).
- There are one or more sets of backup volumes (BKPxxx) that have been created by Db2 Recovery Expert, RBR, or DBR for Db2 backups. The backup volumes are offline and have internal volsers of their corresponding source volumes.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.
- There are two target ICF catalogs (USERCAT.TGT01 and USERCAT.TGT02). It is possible that one of the ICF catalogs resides on a target volume, both ICF catalogs reside on target volumes, or none of the target volumes contain ICF catalogs.

Step 1 - Get the source to backup volume pairing (SRCxxx to BKPxxx) and the source ICF catalog names from the last backup taken for Db2 subsystem DB2P.

Run the ccc#VOLS program to get the source (set A) and backup (set B) volume pairing used by the Db2 Recovery Expert, RBR, or DBR for Db2 backup and the names of the source ICF user catalogs. Sample JCL for this program can be found in the Db2 Recovery Expert, RBR, or DBR for Db2 ScccSAMP library.

Step 2 - Reformat the output of step 1 (backinfo data set) for use in the COPY in steps 3 and 4.

The offline backup volumes are clipped to new volsers and varied online. The user catalog pairs are also specified here. Sample JCL can be found in the installation SCKZJCL library in member CKZBKIRF.

Partial JCL for this step follows:

```
//S1      EXEC  PGM=CKZ00010,REGION=8M
//STEPLIB DD  DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//FRVOLSER DD DSN=HLQ?.WRK.FRVOLSER,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD  DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//VOLOPCMD DD DSN=HLQ?.WRK.VOLOPCMD,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1))
//CKZIN   DD  *
```

```

BACKINFO-REFORMAT -
BACKINFO-DDN(BACKINFO) -
VOLPAIRS-DDN(VOLPAIRS) -
FROM-VOLSER-DDN(FRVOLSER) -
CLIP-IF-OFFLINE(Y) -
VOLSER-RENAME-MASKS(
    SRC* BKP*
) -
VOLOPTIONS-CMD-DDN( VOLOPCMD ) -
USERCATALOGS-DDN(UCATS) -
USERCATALOGS(
    USERCAT.SRC01 USERCAT.TGT01 -
    USERCAT.SRC02 USERCAT.TGT02 -
)
//*

```

Step 3 - Set the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal and back up the source ICF catalogs from the backup volumes.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step when there are target ICF catalogs on the target volumes follows:

```

//S1 EXEC PGM=00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECOG=KS,KEYLEN=64,KEYOFF=0,
// LRECL=600,SPACE=(CYL,(10,10))
//CKZIN DD *
COPY -
DATA-MOVER(PGM(NONE)) -
VOLPAIRS-DDN(VOLPAIRS) -
USERCATALOGS-DDN(UCATS) -
TARGET-UCATS-ON-TARGET-VOLUMES(Y) -
SOURCESONLINE(N) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*

```

Partial JCL for this step when there are no target ICF catalogs on the target volumes follows:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECOG=KS,KEYLEN=64,KEYOFF=0,
// LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECFM=FB,LRECL=80,BLKSIZE=0,
// SPACE=(CYL,(1,1))
//CKZIN DD *
COPY -
DATA-MOVER(PGM(NONE)) -
VOLPAIRS-DDN(VOLPAIRS) -
USERCATALOGS-DDN(UCATS) -
SOURCESONLINE(N) -
CATWORK-DSN(HLQ?.WRK.* ) -
JOURNAL-DDN(JOURNAL)
//*

```

Step 4 - Copy the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step when there are target ICF catalogs on the target volumes follows:

Partial JCL for this step when there are no target ICF catalogs on the target volumes follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//FRVOLSER DD DISP=SHR,DSN=HLQ?.WRK.FRVOLSER
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECOG=KS,KEYLEN=64,KEYOFF=0,
//         LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECFM=FB,LRECL=80,BLKSIZE=0,
//         SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY      -
FROM-VOLSER-DDN(FRVOLSER) -
TO-VOLSER( -
           TGT001 TGT002 -
           ) -
NOUSERCATALOGS -
JOURNAL-DDN(JOURNAL)
//*
```

Step 5 - Run VOLOPTIONS command created in step 2 to vary offline and unclip the backup volumes.

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN   DD DISP=SHR,DSN=HLQ?.WRK.VOLOPCMD
//*
```

Step 6 - Run CKZRNTGT with the VOLPLIST data sets from steps 3 and 4.

Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 3), this step is omitted.

Partial JCL for this step follows:

```
//* CKZIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH A USERCATALOGS KEYWORD,
//* DD VOLPLIST.
//* NUCIN IS FROM THE Db2 Cloning Tool COPY COMMAND WITH THE NOUSERCATALOGS KEYWORD,
//* DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE Db2 Cloning Tool VOLOPTIONS COMMAND
//S2      EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN   DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN   DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT  DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//         DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//         SPACE=(CYL,(1,1))
```

Step 7 - Run VOLOPTIONS with the newtgt data set from step 5 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME.

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

If the TARGET-UCATS-ON-TARGET-VOLUMES(Y) parameter was specified in the first COPY step (Step 3), this step is omitted.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT  DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=SHR
//CKZIN   DD *
          VOLOPTIONS UPDATE           -
          NEWTARGETS-DDN(NEWTGT)      -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 8 - RENAME the data sets on the target volumes (TGTxxx).

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

```
Partial JCL and command:
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD *
          RENAME                       -
          SAFE                          -
          VOLBKUP-DDN(VOLBKUP)          -
          RENAME-MASKS(                 -
              PROD1.** TEST1.**         -
              PROD2.** TEST2.**         -
          )                               -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 9 - Run the Db2 conditioning commands.

Run the Db2 conditioning commands: DB2UPDATE, DB2ALTERBSDS(SLB-START), DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2SCHEMA-UPDATE, DB2STOP, DB2UTILXCLEAN. For additional information about the conditioning commands, refer to the [“Db2 online cloning”](#) on page 134 section for your specific type of cloning.

CKZINI customization values

The CKZINI member (the product initialization member) defines global information regarding Db2 Cloning Tool usage and options within your installation.

This topic provides guidelines, values, and syntax used in the CKZINI member.

Structure of the CKZINI member

The CKZINI member is organized in sections. Each section contains a set of individual parameter specifications known as tokens. A single CKZINI member is recommended for use by all users of Db2 Cloning Tool at the installation site.

Syntax rules

The CKZINI consists of token assignment statements that are organized by sections. The general format for a token statement is:

```
Keyword = Parameter(s)
```

Leading blanks from the beginning of the logical card-image records are allowed for all statements. Syntax scan processing locates the first non-blank character in each logical record. The entire length of the 80-byte logical record is considered for valid data. Do not renumber the CKZINI member; sequence numbers assigned in columns 73 through 80 will cause errors.

You may add a notes token to document your changes to any section including the initial/unnamed section. The INIMERGE program will retain these notes along with any comments that continue from the notes token.

```
Notes = Updated by Dan on 2011/04/23 +  
       Updated by John on 2011/02/20
```

Section names

A *section name* is indicated by the colon character (:), immediately followed by the section name string. Section names are to be changed or added only under the direction of IBM Software Support.

For a multi-image CKZINI, the section name can be qualified by sysplex and/or system name, except for the PRODUCT_INFO and INIMERGE_VALUES sections:

Sysplex-name and system-name qualification:

```
:section-name.sysplex-name.system-name
```

Sysplex-name only qualification:

```
:section-name.sysplex-name
```

System-name only qualification:

```
:section-name.system-name
```

The sysplex and system names must be explicitly specified with no wild-card characters allowed.

INIMERGE will retain your qualified sections and merge new tokens for those sections.

Db2 Cloning Tool will use only the first matching section in the CKZINI member. All other variations of that section will be ignored. You may specify qualified sections for those images with unique requirements, followed by an unqualified section that applies to all of the other images.

Token name

A *token name* is a keyword value that can be specified under particular section names. They are specified as a blank-delimited character string to the left of an equal (=) sign. Token names are to be changed or added only under the direction of IBM Software Support except where noted.

Token value

A *token value* is data that is specified as a set of strings to the right of the equal (=) sign after a token name. Token values may be keywords, user values, or a keyword with a token value, shown as KEYWORD(uservalue).

Values may be enclosed in either single or double quotes. The quotes are stripped away before the product uses the value. Quotes may be supplied as data by:

- Using the opposite quote symbol as delimiters (e.g., " " will yield a single quote as data).
- Specifying two to get one (e.g., ' ' will yield a single quote as data).

Continuation rules

Statements can be continued, using either a minus (-) or plus (+) character anywhere within the text. All data to the right of the continuation character on that logical record is interpreted as a comment and ignored.

Comments

Both line mode and block mode methods of commenting are supported:

- An asterisk (*) or slash-asterisk (/*) in column-1 marks the entire line as a comment. This style of comment is not allowed inside a continued /* */ type comment but is allowed in a continued token/value statement. A line that is entirely blank can also be considered a comment.
- Entire lines, blocks of lines, or portions of a line may be commented by beginning the comment with a slash-asterisk (/*) and terminating the comment with an asterisk-slash (*). Nested comments are honored.

About Db2 Cloning Tool Table Space Cloning token errors

Many tokens in CKZINI have default values. If a token has a default value and there is an error in the token, Db2 Cloning Tool Table Space Cloning substitutes the default value. Db2 Cloning Tool Table Space Cloning then outputs a message describing the error and the default value substitution and produces a return code of four (4).

- If MAX_RC=4, Db2 Cloning Tool Table Space Cloning continues to run to completion.
- If MAX_RC=0, Db2 Cloning Tool Table Space Cloning stops after parameter validation.

CKZINI keyword syntax and descriptions

The CKZINI member keyword syntax and descriptions in this topic help you understand and modify the keywords when necessary.

The CKZINI member parameters are grouped into logical sections.

:PRODUCT_INFO section

| <i>Table 95. Keywords and values for the :PRODUCT_INFO section</i> | |
|--|---|
| Keywords and Values | Description and Usage |
| DB2_CLONING_TOOL_REL = <i>version/release number</i> | Used to verify product version and release. Do not alter. |
| DB2_CLONING_TOOL_REL_DATE = <i>release date</i> | Used to verify product release date. Do not alter. |

:INIMERGE_VALUES section

Important: This section is critical for the INIMERGE process and should not be changed except by authorized IBM Software Support personnel.

Table 96. Keywords and values for the :INIMERGE_VALUES section

| Keywords and Values | Description and Usage |
|---------------------|--|
| SPECIAL_SECTIONS = | <p>The value for this token will be updated only by IBM. For this release, the values are:</p> <pre>SPECIAL_SECTIONS=SI040_VALUES SI027_VALUES</pre> <ul style="list-style-type: none"> • :SI027_VALUES <p>Changes or additions to this section must be authorized by and under the direction of IBM Software Support.</p> <ul style="list-style-type: none"> • :SI040_VALUES <p>Changes or additions to this section must be authorized by and under the direction of IBM Software Support.</p> |

Db2 Cloning Tool Subsystem Cloning sections

The following sections are used for Db2 Cloning Tool Subsystem Cloning.

:DB2_CLONING_TOOL_OPTIONS section

Table 97. Keywords and values for the :DB2_CLONING_TOOL_OPTIONS section

| Keywords and Values | Description and Usage |
|---|--|
| CONCURRENT_EXECUTIONS = Y <u>N</u> | <p>This option addresses the situation where multiple jobs are running concurrently and need exclusive control over the same BCS.</p> <p>Specifying N indicates that if another job has exclusive control over a BCS, the Db2 Cloning Tool job that wants exclusive control of the same BCS will fail with an error.</p> <p>Specifying Y indicates that if another job has exclusive control over a BCS, the Db2 Cloning Tool job that wants exclusive control of the same BCS will wait until the BCS is available. The maximum time to wait is specified with the CONCURRENT_EXECUTIONS_WAIT_TIME keyword.</p> <p>The default is N.</p> |
| CONCURRENT_EXECUTIONS_WAIT_TIME = <i>nnn</i> <u>5</u> | <p>Specifies the maximum wait time in minutes to be used if CONCURRENT_EXECUTIONS = Y is specified. The maximum value that can be specified is 999.</p> <p>The default is 5.</p> |

Table 97. Keywords and values for the :DB2_CLONING_TOOL_OPTIONS section (continued)

| Keywords and Values | Description and Usage |
|--|---|
| SPACE_MANAGEMENT = <i>option</i> | <p>Valid options are: HSM, DMS, ABR(#), or NONE. HSM, DMS, and ABR(#) can be specified as a single option, or, together. NONE may not be specified with any other option.</p> <p>HSM and ABR(#) indicate that Db2 Cloning Tool should use a volume serial of "MIGRAT" as an indication that a data set has been migrated.</p> <p>DMS indicates that Db2 Cloning Tool should use a volume serial of "ARCIVE" as an indication that a data set has been migrated.</p> <p>The default is HSM.</p> |
| SYSPLEX_GROUPNAME_ALL= <i>*ALL</i> <i>sysgrpname</i> | <p>Specifies the scope of the ROUTE command execution. *ALL indicates that the ROUTE command will be run on all systems of the sysplex. The value <i>sysgrpname</i> indicates that the ROUTE command will be run on those systems that are defined in the system group <i>sysgrpname</i>.</p> <p>The default is *ALL.</p> |

:COPY_OPTIONS section

Table 98. Keywords and values for the :COPY_OPTIONS section

| Keyword and Values | Description and Usage |
|--|---|
| CATWORK_ATTR= <i>catalog work data set allocation attributes</i> | <p>Catalog "work" data sets contain catalog entries captured during the COPY step and passed to other steps. This token controls allocation attributes for these data sets if not specified by the COPY CKZIN control statements.</p> <p>Specify attributes in TSO ALLOCATE syntax, e.g., UNIT(SYSALLDA) SPACE(10 10) CYLINDERS.</p> <p>The attributes that can be specified are:</p> <ul style="list-style-type: none"> • DATACLAS(data class name) • MGMTCLAS(management class name) • SPACE(quantity increment) • STORCLAS(storage class name) • TRACKS/CYLINDERS • UNIT(unit) • VOLUME(serial) |

Table 98. Keywords and values for the :COPY_OPTIONS section (continued)

| Keyword and Values | Description and Usage |
|-------------------------------------|--|
| TARGET_VOLS_SHOULD_BE_EMPTY = Y N | <p>Performs a check during the volume pairing process to ensure the target volumes are empty before issuing FlashCopy or SnapShot. Consider the following items:</p> <ul style="list-style-type: none"> • In the event a subsequent RENAME fails and the COPY must be rerun, Db2 Cloning Tool will not clean off the target volumes if "Y" was specified for this parameter. Either initialize the target volumes or change this keyword to "N". • If the "eliminated" target volumes cause there to be more source volumes than targets, the COPY will fail. • If the "eliminated" target volumes still leave at least as many target volumes as source volumes, the pairing will continue as usual. <p>The default is N.</p> |

:DB2_CLONING_TOOL_DEFAULTS section

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section

| Keywords and Values | Description and Usage |
|--|---|
| GDG_ALL_MIGRATED = SKIP RETAIN | <p>This option addresses the situation where a GDG matches a RENAME mask and all of the generations have been migrated.</p> <p>The GDG may be skipped, or the GDS entry may be retained with its new target name.</p> <ul style="list-style-type: none"> • SKIP = BCS update should skip the entry. • RETAIN = the migrated entries should be copied as is to the target BCS. <p>Note: If RETAIN is used, because the migrated generations do not exist under the new name, subsequent access to any generation will fail whether it is accessed specifically or via specification of the base name only. This option is provided to retain relativity.</p> <p>Important: To avoid destroying the relativity of active generations, Db2 Cloning Tool does NOT allow removing selected generations. For data that is migrated and is required on the target volumes, they must be recalled prior to the COPY.</p> <p>The default is SKIP.</p> |
| GDG_ALL_MIGRATED_RETAIN_RC = blank 0 4 | <p>Specifies the return code to be used if GDG_ALL_MIGRATED = RETAIN is specified.</p> |

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

| Keywords and Values | Description and Usage |
|--|---|
| GDG_EMPTY = <u>SKIP</u> RETAIN | <p>This option addresses an empty base GDG that matches a RENAME mask.</p> <p>The GDG entry can be skipped, or the new base entry can be added to the target user catalog.</p> <p>The default is SKIP.</p> |
| GDG_EMPTY_RETAIN_RC = blank 0 4 | <p>Specifies the return code to be used if GDG_EMPTY = RETAIN is specified.</p> |
| GDG_MIGRATED = <u>ERROR</u> RETAIN | <p>This option addresses the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are migrated.</p> <p>The migrated generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.</p> <ul style="list-style-type: none"> • ERROR - BCS update should terminate. • RETAIN - The migrated entry should be copied as is to the target BCS. <p>Note: If RETAINED, because the migrated generation does not exist under the new name, subsequent access to the generation will fail whether it is accessed specifically or via specification of the base name only.</p> <p>To avoid destroying the relativity of active generations, Db2 Cloning Tool does not allow removing selected generations.</p> <p>Retaining non-existent migrated generations may be suitable for situations such as overstated GDG limits (where it is normal for older generations to be migrated and hopefully never accessed), Log Files, etc. where perhaps only the current generation is kept on primary and older migrated generations are kept as a safety factor.</p> <p>The default is ERROR.</p> |
| GDG_MIGRATED_RETAIN_RC = blank 0 4 | <p>Specifies the return code to be used if GDG_MIGRATED = RETAIN is specified.</p> |

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

| Keywords and Values | Description and Usage |
|------------------------------------|---|
| GDG_TAPE = <u>ERROR</u> RETAIN | <p>This option addresses the situation where a GDG matches a RENAME mask and at least one generation is indeed found on a volume, yet one or more generations are on tape.</p> <p>The tape generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.</p> <ul style="list-style-type: none"> • ERROR - BCS update should terminate. • RETAIN - The tape entry should be copied as is to the target BCS. <p>Note: If RETAIN is specified, accessing a target tape GDS will cause a S813 ABEND whether it is accessed specifically or via specification of the base name only.</p> <p>To avoid destroying the relativity of active generations, Db2 Cloning Tool does not allow removing selected generations.</p> <p>Retaining non-existent tape generations may be suitable for situations such as overstated GDG limits where older generation may have been created on tape.</p> <p>The default is ERROR.</p> |
| GDG_TAPE_RETAIN_RC = blank 0 4 | <p>Specifies the return code to be used if GDG_TAPE = RETAIN is specified.</p> |
| ISSUE_CKZ14141I = option | <p>Issues the message CKZ14141I when a data set matches the RENAME-MASKS, but was not on the Db2 Cloning Tool source volumes.</p> <p>You can customize which data set the error reports on using the option(s) specified in the CKZINI member. The DASD, MIG, and TAPE options may be specified in any combination.</p> <ul style="list-style-type: none"> • <u>ALL</u> - Issue message for any data set that matches the RENAME-MASKS, but is not on the source volumes. • DASD - Issue message for any DASD data set that matches the RENAME-MASKS, but is not on the source volumes. • MIG - Issue message for any migrated data set that matches the RENAME-MASKS. • TAPE - Issue message for any tape data set that matches the RENAME-MASKS. • NOMSG - Do not issue message CKZ14141I. <p>The default is ALL.</p> |

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

| Keywords and Values | Description and Usage |
|---|--|
| MAX_RENAME_TASKS = <i>nnn</i> | <p>The <i>nnn</i> value specifies the maximum number of subtasks used by the RENAME command for volume processing if the RENAME CKZIN MAXTASKS is not supplied. At some point, increasing the number of subtasks will cease to increase performance, due to resource contention. Specifying a value that is too large may result in termination due to memory constraints. The maximum allowed value is 255.</p> <p>The default is 5.</p> |
| MISSING_USERCAT_DISP = DELETE <u>KEEP</u> | <p>Specifies the disposition of target volume data sets where the VVDS catalog back-pointer is not a catalog in the list supplied to the COPY step. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is KEEP.</p> |
| MISSING_USERCAT_RC = 0 <u>4</u> 8 | <p>Specifies the return code to be generated for the RENAME command if one or more target volume data sets contain a VVDS catalog back-pointer not in the list supplied to the COPY step. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 4.</p> |
| NOT_RENAMED_DISP = DELETE <u>KEEP</u> | <p>Specifies the disposition of target volume data sets not-renamed because they do not match a rename mask. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is KEEP.</p> |
| NOT_RENAMED_RC = 0 4 <u>8</u> | <p>Specifies the return code to be generated for the RENAME command if one or more target volume data sets are not-renamed because they do not match a rename mask. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 8.</p> |
| ORPHAN_CATENTRY_DISP = DELETE <u>KEEP</u> | <p>Specifies the disposition of target volume data set catalog entries where in some circumstances the data set is not found on the volume. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is KEEP.</p> |
| ORPHAN_CATENTRY_RC = 0 4 <u>8</u> | <p>Specifies the return code to be generated for the RENAME command if one or more target volume data set catalog entries do not have a corresponding volume data set. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 8.</p> |

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

| Keywords and Values | Description and Usage |
|--|--|
| RECATALOG = Y <u>N</u> | <p>Specifies, if "Y", that catalog entries may be replaced if encountered when cataloging target volume data sets. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is N.</p> |
| TEMP_DATASET_DISP = <u>DELETE</u> KEEP | <p>Specifies the disposition of temporary data sets found on target volumes. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is DELETE.</p> |
| TEMP_DATASET_RC = 0 <u>4</u> 8 | <p>Specifies the return code to be generated for the RENAME command if one or more temporary data sets are found on target volumes. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 4.</p> |
| VTOCIX_REBUILDER = <u>MSC</u> /* (MSC OR IBM) */ | <p>Specifies the method for rebuilding the VTOCIX during the RENAME command.</p> <ul style="list-style-type: none"> • IBM - ICKDSF will be used to rebuild the VTOCIX. • MSC- The "on-board" VTOCIX rebuild will be used. <p>For Extended Address Volumes, ICKDSF will always be used to rebuild the VTOCIX.</p> <p>The default is MSC.</p> |

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

| Keywords and Values | Description and Usage |
|--|--|
| <p>RENAME_ERROR = <u>ABORT</u> CONTINUE</p> | <p>This option specifies how processing proceeds when a RENAME error is encountered. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>ABORT will terminate with an RC=8 after the first error to preserve integrity. ABORT is recommended.</p> <p>CONTINUE will continue processing after most errors and the RENAME command will complete with the specified return code unless an error not handled by the CONTINUE logic is encountered.</p> <p> Warning: The use of CONTINUE can cause inconsistencies between the contents of the volumes and catalogs. Possible problems include:</p> <ul style="list-style-type: none"> • Data sets could be cataloged but are not renamed on disk. • Data sets could be renamed on disk but are not cataloged. • Data sets that are not renamed on disk may not be deleted from disk. • GDG base and GDS entries will not exist in the catalog when there is a missing GDS. • A catalog entry may not point at the correct volume, a catalog entry may be invalid. • CONTINUE could leave uncataloged data sets on SMS managed volumes. <p>If this keyword is specified, Db2 Cloning Tool will not guarantee integrity and the given results will not be fixed by Db2 Cloning Tool.</p> <p>The default is ABORT.</p> |
| <p>RENAME_ERROR_CONTINUE_RC = 0 4 <u>8</u></p> | <p>Specifies the return code to be used if RENAME_ERROR = CONTINUE is specified. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 8.</p> |

Table 99. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

| Keywords and Values | Description and Usage |
|--|--|
| <p>ISSUE_VCLOSE = <u>YES</u> NO BEFORE AFTER</p> | <p>Specifies if a catalog modify command will be issued as part of the volume RENAME processing. The catalog modify command is:</p> <pre data-bbox="792 348 1466 401">F CATALOG,VCLOSE(<i>targetvolser</i>)</pre> <p>The Catalog address space (CAS), caches VVDS information. The modify command requests that the VVDS information cached for the target volume be refreshed. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>Possible values for this keyword are:</p> <ul data-bbox="781 669 1451 953" style="list-style-type: none"> • NO specifies the modify command will NOT be issued. • BEFORE specifies that the modify command will be issued only before the VVDS is updated. • AFTER specifies that the modify command will be issued only after the VVDS has been updated. • YES specifies the modify command will be issued both before the VVDS is updated and after the VVDS has been updated. <p>The default is YES.</p> |
| <p>ISSUE_VCLOSE_SCOPE = <u>LOCAL</u> SYSPLEX</p> | <p>If the CKZINI parameter ISSUE_VCLOSE = YES NO BEFORE AFTER is set to YES, BEFORE, OR AFTER, use one of the following values for the ISSUE_VCLOSE_SCOPE keyword:</p> <ul data-bbox="781 1173 1463 1467" style="list-style-type: none"> • LOCAL – The catalog modify command, F CATALOG,VCLOSE(<i>targetvolser</i>) will be issued only on the system that RENAME is running on. • SYSPLEX – The catalog modify command, F CATALOG,VCLOSE(<i>targetvolser</i>), will be issued on the local system, and the modify command will be routed to all the other systems in the Sysplex, via an MVS ROUTE *OTHER command, after the VVDS has been updated. <p>This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is LOCAL.</p> |

:DB2_OPTIONS section

| <i>Table 100. Keywords and values for the :DB2_OPTIONS section</i> | |
|--|--|
| Keyword or Value | Description |
| DB2_XCFCLEAN = <u>Y</u> N Restrictions: This parameter is only used if Db2 data sharing is being used. | Specifies that DB2UPDATE should clean up the target Db2 data sharing group XCF structures and group members. This value is used if the corresponding keyword is not specified in the DB2UPDATE CKZIN control statements. The default is Y. |
| DB2_PLAN = <i>planname</i> | Specifies the plan name that will be used to process SQL statements with Db2. The default is CKZPLAN. |

:RESOURCE_SERIALIZATION section

Installations running CA-MIM/MII with multiple systems and shared DASD need to set the following parameter to "YES" to ensure that when CA-MIM/MII GDIF is inactive, the Db2 Cloning Tool data sets are protected from data corruption.

| <i>Table 101. Keywords and values for the :RESOURCE_SERIALIZATION section</i> | |
|---|--|
| Keyword or Value | Description |
| MIM_GDIF = <u>NO</u> YES | If you have CA-MIM/MII with multiple systems and shared DASD, this token should be changed to YES to ensure that when CA-MIM/MII GDIF is inactive, the Db2 Cloning Tool data sets are protected from data corruption. The default is NO. |

Db2 Cloning Tool Table Space Cloning sections

The following sections are used for Db2 Cloning Tool Table Space Cloning.

:DSN_PRODUCT_PERF section

| <i>Table 102. Keywords and values for the :DSN_PRODUCT_PERF section</i> | |
|---|---|
| Keyword or Value | Description |
| MAX_RC = 0 <u>4</u> | STOP JOB WHEN > MAX_RC . This parm applies to overall processing return codes for a table space cloning procedure. This allows processing to continue when warnings are encountered. The default is 4. |
| MAX_COPY_RC = 0 4 <u>8</u> | STOP JOB WHEN > MAX_COPY_RC OCCURS WHEN COPYING DATA . This parm only applies to copy processing return codes. This allows one or more copies to fail and the others to continue. The default is 8. |

Table 102. Keywords and values for the :DSN_PRODUCT_PERF section (continued)

| Keyword or Value | Description |
|--------------------------------|--|
| MAX_SUBTASKS = <u>1</u> ... 99 | <p>Valid values range from 1 to 99.</p> <p>Db2 Cloning Tool Table Space Cloning uses subtasks to perform several functions, such as catalog access in the source job and SYNCDB2 command processing in the target job. These subtasks allow multiple IOs to be performed concurrently. Changing the number of subtasks Db2 Cloning Tool Table Space Cloning uses may improve performance.</p> <p>Raising the limit of the number of subtasks that can be specified may reduce the elapsed time of the target job. At some point, the larger number of subtasks will place a burden on z/OS and the elapsed time will start to go back up. Each different set of objects may have its own optimal target job value for the number of subtasks.</p> <p>The larger number of subtasks available normally is useful only for the target job. When using more than 18 subtasks in the target job, use a PARMLIB member for the source job and another for the target job. From one to four subtasks is generally sufficient for the source job. The TCPIP server job uses only one subtask, regardless of how many are specified. If the SET command is used, the source and target jobs get the same number of subtasks. This SET value also overrides the value in PARMLIB.</p> <p>The default is 1.</p> |

:TCPIP_OPTIONS section

Table 103. Keywords and values for the :TCPIP_OPTIONS section

| Keyword or Value | Description |
|--|---|
| TCPIP_ENCRYPTION_ENABLE = Y <u>N</u> | <p>If this keyword is set to Y, Db2 Cloning Tool encrypts all network communications between the source and target jobs and the target and source TCP/IP server jobs. The value of TCPIP_ENCRYPTION_ENABLE must be the same on the source and the target systems.</p> <p>Note: This keyword does not control DDF encryption when DDF is used for connecting to the remote Db2 subsystem.</p> <p>The default is N.</p> |
| TCPIP_KEY_LABEL = <i>keylabel</i> | <p>An encryption key label to be used for encrypting TCP/IP network communications. If set, the <i>keylabel</i> value must reference a 256-bit AES encryption key. The Db2 Cloning Tool client and server jobs must use the same encryption key, although it may be referenced by different key labels. If this keyword is not set or is set to an empty value, Db2 Cloning Tool uses the internal encryption key. This keyword has no effect if TCPIP_ENCRYPTION_ENABLE is set to N.</p> |

Table 103. Keywords and values for the :TCPIP_OPTIONS section (continued)

| Keyword or Value | Description |
|---------------------------------|---|
| TCPIP_SERVER_PORT = <u>5099</u> | <p>TCPIP SERVER PORT #</p> <p>This is the port that the Db2 Cloning Tool Table Space Cloning TCP/IP server job listens on for client connects. It must be known to the source job and the TCP/IP server job and must be the same value. It should be an unused number less than 65536.</p> <p>One TCP/IP server only connects to a single Db2 subsystem. If multiple servers are required on a single LPAR image, use a different port number for each server.</p> <p>The default port for this product is 5099.</p> |
| TCPIP_STC_NAME = <u>TCPIP</u> | <p>LOCAL TCPIP STC NAME</p> <p>This defines the name of the TCP/IP started task.</p> <p>TCPIP is the default used by this product, as it is the MVS default.</p> |

:DSN_COPY_OPTIONS section

This section defines the options to be used for the data set copies. These may be overridden on the COPY command for the target.

Table 104. Keywords and values for the :DSN_COPY_OPTIONS section

| Keyword or Value | Description |
|--|---|
| ALWAYS_COPY_INDEXSPACES = <u>Y</u> N | <p>This token determines how index spaces are included in LISTDEFS. When the value is set to Y, all index spaces are included for every table space included in a LISTDEF. No INCLUDE INDEXSPACES syntax is required in the LISTDEF.</p> <p>The default is Y.</p> |
| AUTO_START_SOURCE_SPACE = <u>Y</u> N R | <p>This token allows the source job to optionally start each source Db2 table spaces and index spaces after the copy process is complete.</p> <p>When the value is Y, Db2 Cloning Tool Table Space Cloning starts the source table spaces and index spaces after the copy is complete.</p> <p>When the value is N, source spaces are left stopped after the copy is complete.</p> <p>When the value is R, the source table spaces and index spaces are restored to the status they were before executing the source job.</p> <p>If you set this parm to Y or R, and a table space or index space has a status that is not STOP, RW, RO or STOPP, a RC of eight (8) is issued.</p> <p>The default is Y.</p> |

Table 104. Keywords and values for the :DSN_COPY_OPTIONS section (continued)

| Keyword or Value | Description |
|---|--|
| AUTO_START_TARGET_SPACE = <u>Y</u> <u>N</u> | <p>Db2 Cloning Tool Table Space Cloning will issue a START command for the target space.</p> <p>This token determines if SYNCDB2 processing in the target job issues a DB2 START DATABASE SPACENAM command for the table space or index space being processed after all the Sync IO to the data set is complete. This value determines the value of the START-SPACE command in each invocation of the SYNCDB2 command in the target job.</p> <p>Note: There is one SYNCDB2 command for each target data set in the SYNCDB2 command member.</p> <p>When Db2 Cloning Tool Table Space Cloning builds the SYNCDB2 commands, it uses the value of AUTO-START-TARGET-SPACE from PARMLIB and/or the COPY command to add a similar command to the SYNCDB2 command output member. Note that the COPY command value overrides PARMLIB.</p> <p>The default is Y.</p> |
| AUTO_STOP_TARGET_SPACE = <u>Y</u> <u>N</u> | <p>This parm determines if Db2 Cloning Tool Table Space Cloning issues a STOP DATABASE SPACENAM DB2 command for each target table space and index space. If the value is Y, Db2 Cloning Tool Table Space Cloning will issue a STOP command before the copy begins.</p> <p>Note: If a STOP is issued, it must complete before the copy can begin as Db2 Cloning Tool Table Space Cloning must have exclusive control of the target data set.</p> <p>If the value is N, Db2 Cloning Tool Table Space Cloning assumes the space is already stopped and no Db2 command is issued before the copy begins. If the space is not stopped, an allocation error is issued.</p> <p>The default is Y.</p> |
| COPY_IF_NO_DB2_TARGET_OBJECTS = <u>Y</u> <u>N</u> | <p>Copy even if no Db2 table spaces or index spaces exist on the target.</p> <p>This parm enables Db2 Cloning Tool Table Space Cloning to copy data sets when no target table spaces or index spaces exist. Db2 Cloning Tool Table Space Cloning will use defaults for the high level qualifier (DEFVCAT subcommand from the COPY command) and the fifth level qualifier (F0001) when creating the target data set names.</p> <p>The default is N, which means do not copy the source table spaces or index spaces if no target table spaces or index spaces exist.</p> |
| DB2V11_EPIC37738_INSTALLED = <u>Y</u> <u>N</u> | <p>This parameter affects Db2 Cloning Tool Table Space Cloning behavior for processing versioned objects. Set this parameter to Y if the target Db2 is running Db2 11 and all of the following APARs are installed: PI78780, PI76461, PI76462, PI75145, PI81005, PI76179, and PI80006.</p> <p>The default is N.</p> |

Table 104. Keywords and values for the :DSN_COPY_OPTIONS section (continued)

| Keyword or Value | Description |
|---------------------------------|---|
| DSNS_PER_COPY = 255 | The number of data sets to include in each DSS copy command. Valid range: 1 to 255. A performance benefit may be realized by changing this value. |
| DSS_COPY_COMMANDS = 5 | The number of DSS copy commands in each DSS call. Valid range: 1 to 256. Specifying a large value may result in a storage shortage and S878 abends. |
| NULLSTORCLAS = <u>Y</u> N | Determines the input to the automatic class selection routine (ACS) when copying data sets with ADRDSSU. If set to Y, a null storage class will be passed to the ACS. This allows the ACS routine to assign storage classes to target data sets. If set to N, the input to the ACS will be the source data set's storage class. For more information on this keyword, consult the DFSMSdss Storage Administration documentation. The default is Y. |
| RESET_LOGRBA = <u>Y</u> N A | Reset the LOGRBA in the target table space and index space data sets. This parm determines if SYNCDB2 processing in the target job resets the LOGRBA in the copied table space and index space data sets. This value determines the value of the RESET_LOGRBA command in each invocation of the SYNCDB2 command in the target job. RESET_LOGRBA(A) performs an automatic reset of the LOGRBA depending on the conditions. When the source and target Db2 subsystems are in the same Db2 subsystem, and RESET_LOGRBA is set to A, the LOGRBA is not reset. When the source and target Db2 subsystems are in different Db2 subsystems, and RESET_LOGRBA is set to A, the LOGRBA is reset. Note there is one SYNCDB2 command for each target data set in the SYNCDB2 command member. Failure to specify Y or A may result in the Db2 table spaces and index spaces being unusable after completion of the target job. The default is Y. |

Deallocating target Db2 coupling facility structures

In a data sharing environment, Db2 retains lock information and table information in coupling facility (CF) structures. These structures may exist from a previous running of the target Db2 sharing systems. The target Db2 will use this out-of-date information if it exists during start-up and will end up corrupting tables on the target Db2 subsystem. To prevent this, the target Db2 CF structures need to be deallocated prior to starting the target Db2.

The following steps describe how to determine if there any target Db2 CF structures that need to be deallocated and how to deallocate them.

1. For data sharing, ensure the target Db2 CF structures are deallocated.

The Db2 structure names are:

```
groupname_LOCK1
groupname_SCA
groupname_GBPxxx
```

Where:

- *groupname* is replaced with the name of the Db2 data sharing group.
- GBPxxx is one of these unique identifiers of the buffer pool:

```
GBP0, GBP1, ... GBP49   or
GBP8K0, ... GBP8K9   or
GBP16K0, ... GBP16K9   or
GBP32K, ... GBP32K9
```

2. Issue this z/OS command: **D XCF,STR**
3. Look for any target Db2 structures that are marked as "ALLOCATED".
 - Partial output from D XCF,STR: (*groupname* = DBGT)

```
DBGT_GBP0      --      --      NOT ALLOCATED
DBGT_LOCK1     09/17/2009 10:14:29 ALLOCATED
DBGT_SCA       09/17/2009 10:14:27 ALLOCATED
```

- If *groupname_LOCK1* is marked as "ALLOCATED", issue the following z/OS commands:

```
SETXCF FORCE,CONNECTION,STRNAME=groupname_LOCK1,CONNAME=ALL
SETXCF FORCE,STR,STRNAME=groupname_LOCK1
```

- If *groupname_SCA* is marked as "ALLOCATED", issue the following z/OS command:

```
SETXCF FORCE,STR,STRNAME=groupname_SCA
```

- If any *groupname_GBPxxx* is marked "ALLOCATED", issue the following z/OS commands:

```
SETXCF
FORCE,CONNECTION,STRNAME=groupname_GBPxxx,CONNAME=ALL
SETXCF FORCE,STR,STRNAME=groupname_GBPxxx
```

4. The following command should be issued before starting the target Db2 subsystem: **D XCF,STR**

The command response lines corresponding to the desired Db2 group should show as "NOT ALLOCATED". As an example, you are looking for *groupname* = x where x in this example is the Db2 data sharing group named DBGT. This example is a partial output from the command of only the lines for the Db2 data sharing group (*groupname*) named DBGT.

```
DBGT_GBP0      --      --      NOT ALLOCATED
DBGT_LOCK1     --      --      NOT ALLOCATED
DBGT_SCA       --      --      NOT ALLOCATED
```

Previous changes to Db2 Cloning Tool

This section describes technical changes to Db2 Cloning Tool in previous editions of the user guide.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

2019-03-29

| Description | Related APARs |
|---|---------------|
| <p>The USE-RUNTIME-REPOSITORY functionality was enhanced and the documentation updated to describe restartability of the table space cloning target job. The following topics were added or updated, and messages were updated and added for this enhancement:</p> <ul style="list-style-type: none"> • “Target job overview” on page 207 • “Report job details (optional)” on page 213 • “Runtime repository functionality overview” on page 216 • Chapter 18, “Using image copies to clone table spaces and index spaces,” on page 289 • ISPF interface topics “Table space cloning DD descriptions” on page 330, SET command defaults, and COPY command defaults • “COPY command and keyword definitions” on page 516 • “SET command and keyword definitions” on page 568 | PH09043 |
| <p>The topic “DB2SCHEMA-UPDATE” on page 435 was updated to add details about how the command works.</p> | None |
| <p>Updates to the scenario topics provide additional details for subsystem cloning when ICF catalogs may already reside on the target volumes. The following topics were updated:</p> <ul style="list-style-type: none"> • “Volume cloning using an interim set of volumes” on page 1091 • “Volume cloning using multiple interim sets of volumes” on page 1096 • “Db2 subsystem cloning from a Db2 BACKUP SYSTEM backup” on page 1105 • “Db2 subsystem cloning to a specific point in time from a Db2 BACKUP SYSTEM backup” on page 1115 • “Db2 subsystem cloning from other system level backups when backup volumes are online” on page 1134 • “Db2 subsystem cloning from other system level backups when backup volumes are offline” on page 1139 | None |
| <p>Access to the BPX.FILEATTR.APF FACILITY class profile is no longer required to use log apply functionality with Db2 Cloning Tool Table Space Cloning. The topic “Function authorization requirements” on page 23 was updated.</p> | PH05570 |
| <p>Information was added to “Set up your environment prior to customization” on page 21 to include the requirement for all cloning jobs to be at the same maintenance level.</p> | None |
| <p>The default for TCPIP_ENCRYPTION_ENABLE was changed to N in “CKZINI keyword syntax and descriptions” on page 1146 and text was added stating that this parameter must be set to the same value on the source and the target systems.</p> | None |

2019-01-25

| Description | Related APARs |
|--|---------------|
| <p>Support was added for subsystem and table space cloning when pervasive data encryption in DFSMS and Db2 V12 with function level 502 is implemented. The following topics were updated or added:</p> <ul style="list-style-type: none">• Information was added to the Db2 Cloning Tool Subsystem Cloning topic “Considerations for DFSMS pervasive encryption” on page 111.• The ENCRYPTION-MISMATCH-RC parameter was added to the ISPF interface topic COPY command defaults and the Db2 Cloning Tool Table Space Cloning topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516.• The KEYLABEL parameter was added to the Db2 Cloning Tool Table Space Cloning topic “DDL-ATTRIBUTE-CHANGE parameter values” on page 555.• The TCPIP_ENCRYPTION_ENABLE and TCPIP_KEY_LABEL parameters were added to the CKZINI PARMLIB member as described in “CKZINI keyword syntax and descriptions” on page 1146.• The topic “Db2 subsystem cloning using data set copy” on page 1103 was updated. | PH05821 |
| <p>Support was added to allow you to specify DDF dynamic location aliases for the subsystem cloning DB2UPDATE command. The following topics were modified and messages were updated:</p> <ul style="list-style-type: none">• Stored procedure topic “Parameter files and parameter descriptions” on page 155.• ISPF interface topic “Specifying information for subsystem cloning” on page 325.• Subsystem cloning command topics “DB2UPDATE command syntax” on page 455 and “DB2UPDATE command and keyword definitions” on page 456. | PH05491 |
| <p>Doc clarifications were made to describe the behavior of the table space cloning COPY parameter COPY-IJ-TO-NONEXISTENT-TARGET. The following topics were updated:</p> <ul style="list-style-type: none">• “Considerations for target objects created using DEFINE NO” on page 186.• ISPF interface topic COPY command defaults.• Table space cloning topic “COPY command and keyword definitions” on page 516. | None |

| Description | Related APARs |
|--|---------------|
| <p>Documentation updates include the following:</p> <ul style="list-style-type: none"> • The job names that are generated by Tools Customizer were updated and a section was added that describes the job naming conventions. The topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31 was updated. • The ISSUE-UCAT-UNALLOCATE keyword was missing from the Db2 Cloning Tool Subsystem Cloning RENAME command. The keyword and its parameters were added to “RENAME command syntax” on page 474 and “RENAME command and keyword definitions” on page 477. • Db2 Cloning Tool removes the SLB information from the BSDS as part of Db2 conditioning; the topic “DB2UPDATE” on page 454 was updated with this information. • Information about cloning a mixture of explicitly and implicitly created objects was added to “Implicitly created objects” on page 184. • Many messages were renumbered and updated in the “Db2 Cloning Tool messages” on page 654 reference. | PH06199 |
| <p>The KEEP-VOLUMES-SEQUENCE keyword was incorrect in some topics and has been corrected. The topics “Parameter files and parameter descriptions” on page 155, “Db2 subsystem cloning from a Db2 BACKUP SYSTEM backup” on page 1105, and “Db2 subsystem cloning from Db2 BACKUP SYSTEM dump tapes” on page 1128 were updated.</p> | None |

2018-11-15

| Description | Related APARs |
|---|---------------|
| <p>Job templates were added that can be used to clone consistent FlashCopy data sets from the source system to Db2 data sets on the target system using Sterling Connect:Direct. The following topic were added:</p> <ul style="list-style-type: none"> • “Job templates for cloning consistent FlashCopy data sets to Db2 data sets using IBM Sterling Connect:Direct” on page 307 • “Using the ISPF interface and job templates to clone with IBM Sterling Connect:Direct” on page 309 • “Cloning source VSAM files using Sterling Connect:Direct” on page 310 | PI87947 |
| <p>An Installation Verification Process (IVP) was added for Db2 Cloning Tool Table Space Cloning. The topic “Using the Installation Verification Process” on page 98 was added and the topics “Worksheets: Gathering parameter values for Tools Customizer” on page 31 and “Verifying successful customization” on page 98 were modified.</p> | PH03430 |
| <p>Tools Customizer parameters were added to add additional indexes on the Db2 catalog tables and to provide a buffer pool name for those additional indexes to help improve the performance of the DB2SCHEMA-UPDATE command. The topic “Worksheets: Gathering parameter values for Tools Customizer” on page 31 was updated.</p> | PH04472 |
| <p>The SOURCE-TCPIP-STC-NAME and TCPIP-STC-NAME parameters were clarified in the topic “SET command and keyword definitions” on page 568.</p> | None |

| Description | Related APARs |
|---|---------------|
| <p>The LISTDEF command description was enhanced to describe how COPY command keywords can impact the construction of the LISTDEF command. The topics “LISTDEF” on page 562, “LISTDEF command and keyword definitions” on page 565, and “COPY command and keyword definitions” on page 516 were updated.</p> | PI99297 |
| <p>The topic “How Db2 Cloning Tool Table Space Cloning checks object compatibility” on page 192 was added to clarify how Db2 Cloning Tool Table Space Cloning processes object checking for mismatches.</p> | None |

2018-10-09

| Description | Related APARs |
|---|---------------|
| <p>The KEEP-VOLUMES-SEQUENCE keyword was added to subsystem cloning so that source volumes can be paired with specific target volumes. The following topics were modified and message were added and removed for this feature:</p> <ul style="list-style-type: none"> • “Parameter files and parameter descriptions” on page 155 • “COPY” on page 383, “COPY command syntax” on page 386, and “COPY command and keyword definitions” on page 388 • “RESTORE-FROM-DUMPTAPES command syntax” on page 491 and “RESTORE-FROM-DUMPTAPES command and keyword definitions” on page 491 | PH02084 |
| <p>Subsystem cloning has been enhanced so that you can clone source volumes to target volumes that have ICF catalogs on them, without having to move the target catalogs off the volumes before the cloning and moving them back afterwards. The TARGET-UCATS-ON-TARGET-VOLUME keyword was added to the COPY and RESTORE-FROM-DUMPTAPES commands, and the stored procedure and cloning scenarios were updated for the new keyword. The following topics were added or changed:</p> <ul style="list-style-type: none"> • Chapter 9, “Cloning Db2 subsystems,” on page 125 • “Location of the source and target ICF catalogs” on page 110 • “Parameter files and parameter descriptions” on page 155 • “COPY” on page 383, “COPY command syntax” on page 386, and “COPY command and keyword definitions” on page 388 • “RESTORE-FROM-DUMPTAPES” on page 489, “RESTORE-FROM-DUMPTAPES command syntax” on page 491, and “RESTORE-FROM-DUMPTAPES command and keyword definitions” on page 491 • Cloning scenarios “Db2 subsystem cloning from a Db2 BACKUP SYSTEM backup” on page 1105 and “Db2 subsystem cloning from Db2 BACKUP SYSTEM dump tapes” on page 1128 • Messages were modified and added. | PH01928 |
| <p>The EXECUTE-CREATE-DDL parameter was added to the Db2 Cloning Tool Subsystem Cloning DB2SCHEMA-UPDATE command so that you specify whether to run the DDL as part of the job. The topics “DB2SCHEMA-UPDATE command syntax” on page 436 and “DB2SCHEMA-UPDATE command and keyword definitions” on page 437 were updated.</p> | PH01085 |
| <p>A clarification about using data masking for columns with referential integrity was added to Chapter 17, “Using data masking with table space cloning,” on page 271.</p> | None |

| Description | Related APARs |
|---|---------------|
| Additional job templates CKZJOBT6, CKZJOBT7, and CKZJOBT8 were provided for use when you want to clone table spaces using FUZZY-COPY(Y). The topics Chapter 21, “Using job templates,” on page 303 , Specifying job template data set and member name defaults , and “GLOBAL command values for EMC TimeFinder/Clone Mainframe Snap Facility data set level support” on page 558 were modified. | PH01998 |
| For Db2 subsystem cloning, the following command names were renamed to clarify which volumes are being clipped: <ul style="list-style-type: none"> • CLIP was changed to SOURCECLIP • UNCLIP was changed to TARGETUNCLIP • OFFLINECLIP was changed to TARGETOFFLINECLIP Updates were made throughout the documentation to reflect the command name changes. | None |
| PGM(EMCSNAP) and PGM(SRCVSCPY) were added to some commands and keyword descriptions. The topics “Volume copy products supported by Db2 Cloning Tool” on page 14 , subsystem cloning “COPY command and keyword definitions” on page 388 , and table space cloning “COPY command and keyword definitions” on page 516 were updated. | None |
| The DATA-SHARING keyword of the DB2RBLDBSDS command was clarified in “DB2RBLDBSDS command and keyword definitions” on page 433 . | None |
| The topic “Cloning a Db2 subsystem” on page 112 was updated to add an important note about Db2 address spaces. | None |
| Information was added to the topic “DB2UTILXCLEAN” on page 463 about when to run the command. | None |
| A note was added to BCSCLEAN and COPY-BY-DS to explain how to clean up the target catalog after cloning with COPY-BY-DS. The topics “BCSCLEAN” on page 379 and “Planning for subsystem cloning by data set” on page 404 were updated. | None |
| The BIND-ON-TARGET parameter in the Db2 Cloning Tool Subsystem Cloning stored procedure was updated to describe from where the DB2PLAN value is retrieved. The topic “Parameter files and parameter descriptions” on page 155 was updated. | None |
| The work member name that is updated as part of the DB2SQL command was added to clarify the documentation. The following topics were updated: <ul style="list-style-type: none"> • “Steps for Db2 offline cloning” on page 127 • “Steps for Db2 offline cloning with removal of data sharing members” on page 129 • “Steps for Db2 offline cloning with target becoming non-data sharing” on page 132 • “Steps for Db2 online cloning” on page 137 • “Steps for Db2 online cloning with removal of data sharing members” on page 142 • “Steps for Db2 online cloning with target becoming non-data sharing” on page 148 • “DB2SQL” on page 442 | None |

| Description | Related APARs |
|--|---------------|
| Additional information was provided about the minilog table that is required for log apply processing. The topics “Worksheets: Gathering parameter values for Tools Customizer” on page 31 and Chapter 19, “Using LOG-APPLY to make consistent copies of table spaces and index spaces,” on page 293 were updated. | None |
| The topics “Method 3: Using job templates in the target job” on page 300, “COPY command and keyword definitions” on page 516, and “Report job details (optional)” on page 213 were updated to provide information about several DDs that previously were not described. | None |
| Warning message CKZ23526W was changed to an error message (CKZ23526E); documentation was updated throughout. | None |
| Message “CKZ57508I” on page 951 was added. | PH02002 |
| Message “CKZ11157E” on page 780 was modified. | None |

2018-08-29

| Description | Related APARs |
|---|---------------|
| A set of sample members are provided to help you create a table space cloning workflow that can be used with the z/OSMF REST API. The samples include templates, configuration and input files, and instructions on how to modify the files and use these files in a workflow. This solution allows Db2 for z/OS to participate in DevOps provisioning via REST calls. The full set of topics are provided in the chapter Chapter 24, “Table space cloning using z/OSMF REST services ,” on page 357. | PH00169 |
| A restriction was removed for intelligent rebuild of indexes that required indexes to be cloned with their table spaces. The topic “Method 1: Intelligent rebuild” on page 299 was updated and messages were added. | PH00626 |
| A note was modified to clarify when Db2 Cloning Tool Table Space Cloning can copy table spaces and index spaces between different versions of Db2. The topic “Subsystem copy rules” on page 183 was modified. | None |
| You can now use multi-factor authentication for Db2 Cloning Tool Table Space Cloning with DDF connections, and use RACF password phrases for tasks that are added to the Db2 administrative task scheduler in the Db2 Cloning Tool Subsystem Cloning stored procedure. The topics “Parameter files and parameter descriptions” on page 155 and “COPY command and keyword definitions” on page 516 were updated. | PH00828 |

2018-07-16

| Description | Related APARs |
|---|---------------|
| The CKZD DD is required when using the ISPF interface for Db2 Cloning Tool Table Space Cloning. The topic “Table space cloning DD descriptions” on page 330 was updated. | None |
| The PROCESS-TYPE(A) parameter description of the Db2 Cloning Tool Table Space Cloning COPY command was updated. The topic “COPY command and keyword definitions” on page 516 was updated and message CKZ54401I was added. | PI98688 |

| Description | Related APARs |
|--|---------------|
| <p>Enhancements were made to how Db2 Cloning Tool Table Space Cloning processes indexes. If the source job detects that an index should be rebuilt, its data sets are excluded from the cloning in the source job and the index is rebuilt in the target job. The following topics were updated and messages were added or changed:</p> <ul style="list-style-type: none"> • Chapter 20, “Options for rebuilding indexes,” on page 297 • “Method 1: Intelligent rebuild” on page 299 • COPY command defaults • “COPY command and keyword definitions” on page 516 | PI99726 |
| <p>LOG-APPLY functionality requires READ access to FACILITY class profile BPX.FILEATTR.APF. The topic “Function authorization requirements” on page 23 was updated.</p> | None |
| <p>A recommendation that the CHECK-DATASET-COMPATIBILITY keyword to be set to Y for an initial simulation table space cloning run to ensure source and target data set compatibility was added. The following topics were updated:</p> <ul style="list-style-type: none"> • Source job scenarios were updated for all four table space cloning scenario types in the following topics: “Step 2: Set up and execute source job (CKZ1SRC)” on page 220, “Step 2: Set up and execute source job (CKZ2SRC)” on page 227, “Step 2: Set up and execute source job (CKZ3SRC)” on page 238, and “Step 2: Set up source job (CKZ4SRC)” on page 248. • Table space cloning ISPF interface topic COPY command defaults. • Table space cloning command reference topic “COPY command and keyword definitions” on page 516. | None |
| <p>Updates and corrections to documentation were done for the following topics. In addition, messages were added, updated, and removed:</p> <ul style="list-style-type: none"> • Source job scenarios were updated for all four table space cloning scenario types in the following topics: “Step 2: Set up and execute source job (CKZ1SRC)” on page 220, “Step 2: Set up and execute source job (CKZ2SRC)” on page 227, “Step 2: Set up and execute source job (CKZ3SRC)” on page 238, and “Step 2: Set up source job (CKZ4SRC)” on page 248. • Job templates topics: Chapter 21, “Using job templates,” on page 303, “Job templates for utilities” on page 306, “Predefined processing variables” on page 303, and “Data set processing variables” on page 305. • Table space cloning ISPF interface topic SET command defaults. • Table space cloning command reference: “COPY command syntax” on page 514, “COPY command and keyword definitions” on page 516, “SET command syntax” on page 567, and “SET command and keyword definitions” on page 568. • “CKZINI keyword syntax and descriptions” on page 1146. | None |

| Description | Related APARs |
|---|---------------|
| <p>Db2 Cloning Tool Table Space Cloning now allows both DNS names and IP addresses to be used when specifying TCP/IP configurations. The following topics were updated, and messages were added or updated for this enhancement:</p> <ul style="list-style-type: none"> • “TCP/IP server job details (optional)” on page 211 • “Using LOG-APPLY across multiple LPARs” on page 294 • “COPY command syntax” on page 514 • “COPY command and keyword definitions” on page 516 • “SET command syntax” on page 567 • “SET command and keyword definitions” on page 568 | PI97051 |
| <p>Enhancements were made to Db2 Cloning Tool Table Space Cloning INTELLIGENT-REBUILD and REBUILD-INDEXES-EXECUTE processing to allow you to generate REBUILD statements without running them. The following topics were modified and messages were added:</p> <ul style="list-style-type: none"> • Chapter 20, “Options for rebuilding indexes,” on page 297 • “Recommendations when rebuilding indexes” on page 298 • “Method 1: Intelligent rebuild” on page 299 • “Method 2: Using job templates in the source job” on page 300 • “Method 3: Using job templates in the target job” on page 300 • SET command defaults • “SET command syntax” on page 567 • “SET command and keyword definitions” on page 568 • The topic "Method 1: Dynamic rebuild" was removed. | PI98101 |
| <p>The topic “Set up your environment prior to customization” on page 21 was modified to update the z/OS level to 2.1 or later.</p> | None |
| <p>REUSE and SMS classes keywords were added to Db2 Cloning Tool Table Space Cloning to allow you to allocate new and re-allocate existing VSAM data sets with specified SMS classes during the cloning. The following topics were added or updated and messages were added:</p> <ul style="list-style-type: none"> • COPY command defaults • Specifying DATA-MOVER PGM defaults • “COPY command syntax” on page 514 • “COPY command and keyword definitions” on page 516 | PI97653 |

2018-05-10

Change descriptions

| Description | Related APARs |
|--|---------------|
| <p>Db2 Cloning Tool messages were updated in the topic “Messages” on page 583.</p> | None |

| Description | Related APARs |
|---|---------------|
| <p>Using improved masking capabilities, Db2 Cloning Tool Table Space Cloning now provides DDL generation and execution for all of the following types of objects: storage groups, databases, table spaces, tables, indexes, views, aliases, synonyms, triggers, stored procedures, functions, data types, sequences, sequence aliases, global variables, and materialized query tables. The following topics were updated:</p> <ul style="list-style-type: none"> • “Considerations for Db2 Version 9.1 and later” on page 200 • “Considerations for generating target object DDL using PROCESS-DDL” on page 195 • “Sample DDL processing scenarios” on page 198 • ISPF interface topics: <ul style="list-style-type: none"> – COPY command defaults – Specifying DDL attribute change defaults • “COPY command syntax” on page 514 • “COPY command and keyword definitions” on page 516 • “DDL-ATTRIBUTE-CHANGE parameter values” on page 555 <p>As part of this enhancement, generated DDL is always written to the DD name that is identified in the PROCESS-DDL-DDN(<i>ddname</i>) parameter. Therefore, PROCESS-DDL-DDN is now required unless PROCESS-TYPE is N and no objects with identity columns are cloned. You should ensure that your existing source jobs that enable DDL generation or that clone objects with identity columns include the PROCESS-DDL-DDN parameter.</p> | PI92866 |
| <p>If you plan to use fast replication via the DFSMSdss (ADRDSSU) COPY utility, review the links to DFSMS Advanced Copy Services and DFSMSdss Storage Administration documentation that are in the topic “Verify that your environment meets software requirements” on page 21 prior to performing a cloning.</p> | None |

| Description | Related APARs |
|---|---------------|
| <p>IBM LISTDEF functionality is now implemented for the Db2 Cloning Tool Table Space Cloning LISTDEF command. This enhancement improves source job storage usage during LISTDEF processing. Most of this enhancement is transparent, but some commands and messages were added, updated, or removed to accommodate this change.</p> <p>The following topics were updated:</p> <ul style="list-style-type: none"> • “Set up your environment prior to customization” on page 21 • “Considerations for using catalog prefetch to populate the object cache” on page 201 • ISPF interface topics: <ul style="list-style-type: none"> – COPY command defaults – Specifying target CATALOG-PREFETCH databases • “COPY command syntax” on page 514 • “COPY command and keyword definitions” on page 516 • “LISTDEF” on page 562 • “LISTDEF command syntax” on page 564 • “LISTDEF command and keyword definitions” on page 565 • “SET command and keyword definitions” on page 568 • Many messages were added, updated, and removed from the topic “Messages” on page 583. | PI92862 |

2018-03-14

Change descriptions

| Description | Related APARs |
|--|---------------|
| <p>A note was added to subsystem cloning topics that describes conditions for DB2 11 that must be met prior to the target Db2 becoming non-data sharing. The topics “Db2 offline cloning with target becoming non-data sharing procedure” on page 131 and “Db2 online cloning with target becoming non-data sharing procedure” on page 146 were updated.</p> | None |

| Description | Related APARs |
|---|---------------|
| <p>Versioning support was enhanced for Db2 Cloning Tool Table Space Cloning to facilitate copying objects that have active versioning on source and/or target Db2 subsystems. The following topics were updated:</p> <ul style="list-style-type: none"> • The topic “Copying versioned objects” on page 204 was updated. • TB_VERSION was added as a mismatch type for OBJECT-MISMATCH-RETURN-CODE in “COPY command and keyword definitions” on page 516. • This enhancement deprecates the COPY command parameter WARN-ON-VERSIONS. The parameter was removed from the following topics: <ul style="list-style-type: none"> – COPY command defaults – “COPY command syntax” on page 514 – “COPY command and keyword definitions” on page 516 • The DB2V11_EPIC37738_INSTALLED keyword was added to the CKZINI parameter file. The topic “CKZINI keyword syntax and descriptions” on page 1146 was updated. • Messages were added and several messages were removed in “Messages” on page 583. | PI93478 |
| <p>Data set requirements were added for the CKZERROR, CKZPRINT, and CKZLOG DDs for Db2 Cloning Tool Table Space Cloning. The following topics were updated:</p> <ul style="list-style-type: none"> • “Message output” on page 191 • “Table space cloning DD descriptions” on page 330 • “Step 2: Set up and execute source job (CKZ1SRC)” on page 220 • “Step 3: Set up and execute target job (CKZTRG)” on page 223 • “Step 2: Set up and execute source job (CKZ2SRC)” on page 227 • “Step 6: Set up and execute target job (CKZTRG)” on page 233 • “Step 2: Set up and execute source job (CKZ3SRC)” on page 238 • “Step 7: Set up and execute target job (CKZTRG)” on page 244 • “Step 2: Set up source job (CKZ4SRC)” on page 248 • “Step 10: Set up and execute target job (CKZTRG)” on page 256 | None |
| <p>A note was added to the DATA-MOVER (ADRDSSU) command descriptions to explain the addition of the VOLCOUNT(ANY) and TGTALLOC(SRC) parameters to the DFSMSdss COPY command. The topics COPY command defaults in the ISPF interface topics and “COPY command and keyword definitions” on page 516 were updated.</p> | PI91126 |
| <p>Keywords were added to the subsystem cloning DB2START command in the ISPF interface to allow you to specify the action to take when waiting for Db2 to start or when Db2 prematurely stops during cloning. The topic DB2START command defaults was updated.</p> | PI91925 |
| <p>The CREATOR parameter of the table space cloning COPY command OBJECT-TRANSLATE keyword was revised in the topic “COPY command and keyword definitions” on page 516.</p> | None |

| Description | Related APARs |
|--|----------------|
| <p>When cloning table spaces, data set allocation of target data sets was enhanced. The NULLSTORCLAS parameter for DATA-MOVER PGM(ADRDUSSU) was changed from an optional parameter to a Y/N parameter that defaults to Y to allow the storage class for the target data sets to be determined by ACS routines. The topics “COPY command syntax” on page 514 and “COPY command and keyword definitions” on page 516 were updated. The parameter was added to the CKZINI member as described in “CKZINI keyword syntax and descriptions” on page 1146, and to the Tools Customizer Product Parameters panel that is described in “Worksheets: Gathering parameter values for Tools Customizer” on page 31.</p> | <p>PI91924</p> |
| <p>Additional object mismatches were added to the OBJECT-MISMATCH-RETURN-CODE keyword of the Db2 Cloning Tool Table Space Cloning copy command. The topic “COPY command and keyword definitions” on page 516 was updated.</p> | <p>PI91442</p> |
| <p>The Db2 Cloning Tool Table Space Cloning LOG-APPLY parameter SPACES-PER-MINILOG in the topic “COPY command and keyword definitions” on page 516 was updated.</p> | <p>None</p> |
| <p>The Db2 Cloning Tool Table Space Cloning RI parameter in the topic “LISTDEF command and keyword definitions” on page 565 was updated to clarify that the default is determined by the INCLUDE-ALL-RI parameter in the COPY command.</p> | <p>None</p> |

| Description | Related APARs |
|---|------------------|
| <p>The Db2 Cloning Tool Subsystem Cloning command DB2SCHEMA-UPDATE generates and runs the SQL statements that are necessary to re-create some types of objects that cannot be processed by the Db2 CATMAINT utility with new schema values, and invokes the Db2 CATMAINT utility to change schema values for the rest of the objects. The following topics were added or updated:</p> <ul style="list-style-type: none"> • “Worksheets: Gathering parameter values for Tools Customizer” on page 31 • “Cloning a Db2 subsystem” on page 112 • All of the step tables and procedure descriptions in the topic Chapter 9, “Cloning Db2 subsystems,” on page 125 were updated to add the optional DB2SCHEMA-UPDATE step. • The SCHEMA-MASKS parameter was added the stored procedure parameters. The topic “Parameter files and parameter descriptions” on page 155 was updated. In addition, all stored procedure examples in the topic Chapter 10, “Subsystem cloning using the Db2 Cloning Tool stored procedure,” on page 153 were updated with the new parameter. • The “Subsystem cloning job reference” on page 341 for the ISPF interface was updated. • The following topics were added to the subsystem cloning command reference: <ul style="list-style-type: none"> – “DB2SCHEMA-UPDATE” on page 435 – “DB2SCHEMA-UPDATE command syntax” on page 436 – “DB2SCHEMA-UPDATE command and keyword definitions” on page 437 – “DB2SCHEMA-UPDATE step JCL example” on page 439 • A note that states that certain commands must use the same value for DB2-NAME was modified throughout the user guide to include DB2SCHEMA-UPDATE as one of those commands. • The cloning scenarios and examples in the topic “Cloning scenarios” on page 1091 were updated to add DB2SCHEMA-UPDATE as a conditioning command. • Messages were added to “Messages” on page 583. | PI85474 |
| <p>Data masking functionality requires Unix System Services. The topics “Set up your environment prior to customization” on page 21 and Chapter 17, “Using data masking with table space cloning,” on page 271 were updated to describe those requirements.</p> | None |
| <p>Improved handling of objects with version mismatches was added to Db2 Cloning Tool Table Space Cloning. The following topics were updated:</p> <ul style="list-style-type: none"> • “Copying versioned objects” on page 204 • COPY command defaults • “COPY command and keyword definitions” on page 516 • Messages were added and updated in “Messages” on page 583. | PI89824, PI90127 |

| Description | Related APARs |
|--|---------------|
| <p>When cloning XML table spaces in environments where the ZPARM setting XML_RANDOMIZE_DOCID is set to NO, the target now is updated with a unique XML DOCID. In addition, the SET keyword UPDATE-DOCID-JCL-DSN was added so that Db2 Cloning Tool Table Space Cloning can generate additional jobs to update the DOCID sequence number when the target is a member of a data sharing group. The following topics were updated:</p> <ul style="list-style-type: none"> • “XML considerations” on page 185 • “SET command syntax” on page 567 • “SET command and keyword definitions” on page 568 • SET command defaults • Messages were added to “Messages” on page 583. | PI83216 |
| <p>Support was added for XML table spaces when cloning using LOG-APPLY. As part of that change, the XMLSTRING DD is no longer required. The following topics were updated:</p> <ul style="list-style-type: none"> • “XML considerations” on page 185 • Chapter 19, “Using LOG-APPLY to make consistent copies of table spaces and index spaces,” on page 293 • “Table space cloning steps summary” on page 350 • “Verify the source TCP/IP server job settings” on page 353 • “Build the table space cloning jobs from a profile” on page 353 • “COPY command syntax” on page 514 • “COPY command and keyword definitions” on page 516 • “SET command syntax” on page 567 • “SET command and keyword definitions” on page 568 • Messages were added, revised, and removed in “Messages” on page 583. | PI83216 |
| <p>Changes were made to ALTER TABLE processing when SIM(A) is specified during table space cloning. The topic “Identity columns” on page 184 was updated.</p> | PI87592 |

2017-09-08 edition

| Description | Related APARs |
|---|---------------|
| <p>Table space and index cloning now can be accomplished by copying data sets from the source Db2 subsystem to the target Db2 subsystem using VSAM to VSAM copy. No fast replication or image copies are required. The following topics were updated:</p> <ul style="list-style-type: none"> • “How Db2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 190 • COPY command defaults • “COPY” on page 513 • “COPY command syntax” on page 514 • “COPY command and keyword definitions” on page 516 • Messages were added and revised in “Messages” on page 583. | PI82463 |

SC27-6556-02, 2017-07-25

New information was added for Version 3, Release 2. The documentation changes for this release are listed in this section.

• **Installation and configuration updates**

- The Allocate Table Space Cloning data sets step was updated to add four new data sets for the UNLOAD/LOAD feature that are created when this step is selected. The topic [“Worksheets: Gathering parameter values for Tools Customizer” on page 31](#) was updated.

• **Db2 Cloning Tool Subsystem Cloning**

- An option was added to the CKZDUTST REXX exec to select which objects in UT* status have the UT status removed. The topics [“DB2UTILXCLEAN” on page 463](#) and [“Step JCL example to identify objects with UT status” on page 465](#) were updated.

• **Db2 Cloning Tool Table Space Cloning**

- Enhancements were made to the process for rebuilding indexes, including several new keywords that offer more flexibility and more efficient resource utilization when rebuilding indexes. The topics [“SET command syntax” on page 567](#), [“SET command and keyword definitions” on page 568](#), and [SET command defaults](#) were updated. The topic [Chapter 20, “Options for rebuilding indexes,” on page 297](#) was revised to include the new methods and offer recommendations for rebuilding indexes.
- You can now copy real-time and RUNSTATS statistics from source objects to target objects as part of the cloning process. The topics [“COPY command syntax” on page 514](#), [“COPY command and keyword definitions” on page 516](#), and [COPY command defaults](#) were updated.
- The GMT-OFFSET keyword was added, which can be used to set a specific GMT offset for the TIMESTAMP field during LOG-APPLY processing. The topics [Specifying LOG-APPLY defaults](#), [“COPY command syntax” on page 514](#), and [“COPY command and keyword definitions” on page 516](#) were updated. Messages were added and updated for this feature.
- The USE-LOCAL-TIME keyword for LOG-APPLY was added to the topics [“COPY command syntax” on page 514](#), and [“COPY command and keyword definitions” on page 516](#).
- The SKIP-LOG-APPLY parameter was added to the COPY command and the ISPF interface so you can omit the log read and log apply process when cloning from image copies. The topic [“Selecting the last image copy to use as the cloning source” on page 292](#) was added. The topics [Specifying LOG-APPLY defaults](#), [“COPY command syntax” on page 514](#), and [“COPY command and keyword definitions” on page 516](#) were modified and messages were added.
- The UNLOAD-LOAD keyword was added to the COPY command. This keyword provides an option for the table space cloning process to unload and load table spaces that are ineligible for copy processing when certain types of object mismatches are encountered. The following topics were updated and messages were added or updated for this feature:
 - [“Object attributes that must be identical between source and target” on page 192](#)
 - [“Table space cloning DD descriptions” on page 330](#)
 - [COPY command defaults](#)
 - [Specifying UNLOAD-LOAD defaults](#)
 - [“COPY command syntax” on page 514](#)
 - [“COPY command and keyword definitions” on page 516](#)
- The USE-LAST-CONSISTENT-FLASHCOPY command was added so that you can use data sets that have been created using a Db2 COPY utility with SHRLEVEL CHANGE and FLASHCOPY CONSISTENT options as the source for table space cloning. The topics [“COPY command syntax” on page 514](#), [“COPY command and keyword definitions” on page 516](#), and [COPY command defaults](#) were updated and messages were added.
- The AUTO-TABLESPACE-TRANSLATE and AUTO-INDEXSPACE-TRANSLATE keywords were added. When specified, Db2 Cloning Tool Table Space Cloning attempts to match the names of explicitly created table spaces and index spaces between source and target tables. The topics [“COPY](#)

[command syntax](#)” on page 514, [“COPY command and keyword definitions”](#) on page 516, and [COPY command defaults](#) were updated and messages were added.

- Db2 Cloning Tool Table Space Cloning no longer requires that you create a table to be used for XML processing. The topic [Setting table space cloning defaults](#) was updated. The topic that describes how to define the XML object using the ISPF interface was removed, and the topic [“XML considerations”](#) on page 185 was revised.
- The SLOUT DD is no longer required for processing ALTER TABLE SQL for identity columns. The topic [“Identity columns”](#) on page 184 was updated. JCL samples that showed the SLOUT DD were updated throughout. The SLOUT DD description was removed from the topics [“COPY command syntax”](#) on page 514, [“COPY command and keyword definitions”](#) on page 516, [“SET command syntax”](#) on page 567, and [“SET command and keyword definitions”](#) on page 568. Several messages were removed.
- The &&PARTNUM variable was added to job templates that can be used for a table space or index space PART number. The topic [“Job templates for utilities”](#) on page 306 was updated.
- Improved handling of implicitly created target objects allows you to clone implicit LOB and XML table spaces without the need for the providing matching source-to-target names via the OBJECT-TRANSLATE command. The topic [“Implicitly created objects”](#) on page 184 was added and messages were added for this feature.
- The default value of the DSS_COPY_COMMANDS CKZINI keyword was changed to 5. The topics [“CKZINI keyword syntax and descriptions”](#) on page 1146 and [“COPY command and keyword definitions”](#) on page 516 were updated.
- Default values were changed for the following commands and PARMLIB parameters: MAX-COPY-RC, MAX-RC, REBUILD-INDEXES-EXECUTE, TARGET-JOB-REPAIR-EXECUTE, TARGET-JOB-REPAIR-SELECT, SIMULATE, LONGVAR-COMPATIBILITY, WARN-ON-VERSIONS, and ALWAYS-COPY-INDEXSPACES. The topics [“COPY command syntax”](#) on page 514, [“COPY command and keyword definitions”](#) on page 516, [“SET command syntax”](#) on page 567, and [“SET command and keyword definitions”](#) on page 568 were updated and messages were updated or added.
- Data masking restrictions were updated to specify that data masking cannot be performed on columns that are defined as part of a hash key. The following topic was updated: [Chapter 17, “Using data masking with table space cloning,”](#) on page 271.
- The SET command TCPIP-SERVER-IP keyword was missing and has been added. The topics [“SET command syntax”](#) on page 567 and [“SET command and keyword definitions”](#) on page 568 were updated.

• Changes to the ISPF interface

- A batch import and export utility was added to Db2 Cloning Tool. This utility allows you to export and import data from ISPF cloning profiles and Db2 subsystem information from the VSAM clone profile repository and Db2 control file to and from external files. The topics [“Exporting and importing ISPF cloning profiles and subsystem information”](#) on page 99, [“EXPORT command syntax”](#) on page 100, and [“IMPORT command syntax”](#) on page 101 were added. Messages were also added for this feature.
- You can now apply LOG-APPLY settings to multiple table space cloning profiles on the **DB2 Tablespace Clone Profile Display**. In addition, you can create synchronization jobs to clone the objects spread across multiple table space cloning profiles to the same TO_LOGPOINT. The topic [“Table space cloning across multiple profiles”](#) on page 355 was added, and a message was added for this function.
- The USE-TCPIP field was removed from the **DB2 tablespace LOG-APPLY Command** panels. This value is set automatically by Db2 Cloning Tool Table Space Cloning. The topic [Specifying LOG-APPLY defaults](#) was updated.
- The EXCLUDE-MISMATCH-PROCESSING command was added to the **DB2 tablespace clone COPY Command** panel. The topic [COPY command defaults](#) was updated.

SC27-6556-01, January 2017

New information has been added for Version 3, Release 2. The documentation changes for this release are listed in this section.

• Installation and configuration updates

- Support for Db2 V12 was added. The topic [“Set up your environment prior to customization” on page 21](#) was updated.
- Updated the Db2 version support sentence in the [“Set up your environment prior to customization” on page 21](#) topic to clarify that Db2 Cloning Tool is only supported on a Db2 version through end of support (EOS) for that Db2 version.

• Db2 Cloning Tool Subsystem Cloning

- Support was added for cloning from a system-level backup (SLB) that was created by using a HSM FRBACKUP command. The topics [Chapter 10, “Subsystem cloning using the Db2 Cloning Tool stored procedure,” on page 153](#), [“Parameter files and parameter descriptions” on page 155](#), [“DB2GETBACKINFO” on page 424](#), [“DB2GETBACKINFO command syntax” on page 425](#), and [“DB2GETBACKINFO command and keyword definitions” on page 426](#) were updated. Messages were added and updated for this feature.
- The MIGRATED-DSN parameter was added to the COPY-BY-DS command to allow you to specify how migrated data sets can be handled. The topics [“Planning for subsystem cloning by data set” on page 404](#), [“COPY-BY-DS command syntax” on page 407](#), and [“COPY-BY-DS command and keyword definitions” on page 407](#) were updated. Messages were added and updated for this feature.
- A conditional restart job for subsystem cloning from a system-level backup was added to the ISPF interface. The topic [“Subsystem cloning job reference” on page 341](#) was updated.
- For Db2 V12, the subsystem cloning process was updated to handle cloning of a Db2 system that has a compressed Db2 directory LOB (DSNDB01.SYSDBDXA). The topics [“DB2SQL command syntax” on page 443](#) and [“DB2SQL command and keyword definitions” on page 444](#) were updated and messages were added for this feature. Note that Db2 APAR PI67551 must be applied for this enhancement.
- The DB2UPDATE STOGROUPS keyword was updated to allow wildcards to be used in the source and target entries. The topic [“STOGROUPS keyword considerations” on page 456](#) was added, and the topics [“DB2UPDATE command syntax” on page 455](#) and [“DB2UPDATE command and keyword definitions” on page 456](#) were updated. Messages were added and changed for this feature.

• Db2 Cloning Tool Table Space Cloning

- For Db2 V12, several new object mismatch types can be processed by the COPY command. The topics [“COPY command and keyword definitions” on page 516](#) and [“Object attributes that must be identical between source and target” on page 192](#) were updated and messages were added. The ISPF interface was also updated to incorporate the new mismatch types.
- The topic [Chapter 20, “Options for rebuilding indexes,” on page 297](#) was updated to describe the steps to use two methods of rebuilding indexes concurrently.
- The REBUILD-INDEXES-EXECUTE keyword description was modified to remove the requirement to specify a DD for TARGET-JOB-INDEX-REBUILD-DDN. The topics [SET command defaults](#) and [“SET command and keyword definitions” on page 568](#) were updated.
- Several new keywords were added to the LOG-APPLY command to allow you to better control the log apply process. The NUMBER-OF-SORTS, USE-QUIESCE-POINT-FOR-LOGPOINT, IMAGE-COPY-PREFERENCE, and END-POINT parameters TO_TIMESTAMP, TO_LOGPOINT and TO_QUIESCE were added to command syntax and to the ISPF interface. The following topics were updated and messages were added or updated to support these changes:
 - [“COPY command syntax” on page 514](#)
 - [“COPY command and keyword definitions” on page 516](#)
 - [“Specifying information for table space cloning” on page 325](#)
 - [Specifying LOG-APPLY defaults](#)

- Changes to the ISPF interface were made to allow you to specify a group attach name for table space cloning. The topics [“Specifying information for table space cloning”](#) on page 325 and [“Select the source and target Db2 subsystems”](#) on page 350 were updated and messages were updated to support these changes.
- Changes to the ISPF interface were made so that you can specify common TO_LOGPOINT and other LOG-APPLY parameters for multiple profiles. Messages were added for this enhancement.
- Updates were made to object types in tables for the following topics: [Specifying DDL attribute change defaults](#) and [“DDL-ATTRIBUTE-CHANGE parameter values”](#) on page 555.

SC27-6556-00, August 2015

New information was added for Version 3, Release 2. The documentation changes for this release are listed in this section.

• Installation and configuration updates

- The following updates were made to [“Worksheets: Gathering parameter values for Tools Customizer”](#) on page 31:
 - Some of the job names and template names changed and were updated.
 - On the **DB2 Parameters** panel:
 - The **Plan or Package Owner** parameter was added to allow you to optionally specify the owner of plan and package binds for Db2 Cloning Tool.
 - Two Db2 Cloning Tool-specific parameters were replaced with Tools Customizer parameters. The **System ID where this Db2 normally runs** parameter was replaced with the **SYSAFF for DB2 utilities** parameter, and the **DB2 SQLID for object creates** parameter was replaced with the **SQL authorization ID** parameter.
 - On the Product Parameters panel:
 - The Run CKZINI update step and its parameters were added to create a job that updates the CKZINI member with new values provided in this release.
 - The Db2 plan binds have been moved to their own task and step.
 - Several steps were added to drop and define the global temporary table required for the Db2 Cloning Tool Subsystem Cloning stored procedure.
 - A new step to allocate the ISPF interface CLIST data set was added under the ISPF interface task.
- The instructions in the topic [“Adding Db2 Cloning Tool to the Db2 Admin Launchpad”](#) on page 99 were modified.

• Db2 Cloning Tool Subsystem Cloning

- The COPY-BY-DS command was added. This optional command clones a Db2 subsystem at the data set level by copying a list of data sets, as determined by RENAME masks, to target data sets with new high level qualifiers. The following topics were added or modified, and messages were added to support this feature:
 - [“COPY-BY-DS”](#) on page 404
 - [“COPY-BY-DS command syntax”](#) on page 407
 - [“COPY-BY-DS command and keyword definitions”](#) on page 407
 - [“COPY-BY-DS step JCL example”](#) on page 411
 - [“Db2 subsystem cloning using data set copy”](#) on page 1103
 - [“DB2SQL”](#) on page 442
 - [“DB2UPDATE”](#) on page 454
 - [“Function authorization requirements”](#) on page 23
 - [Chapter 10, “Subsystem cloning using the Db2 Cloning Tool stored procedure,”](#) on page 153

- A change was made to the stored procedure that can be used for cloning Db2 subsystems. The functionality of the stored procedure remains the same; however, the stored procedure calling parameter list was modified and the modules have changed. In addition, new parameters were added to the parameter list for the COPY-BY-DS command and topics were updated to explain the user ID and password requirements for the stored procedure. The following topics were updated or added, and messages were added or updated for this feature.
 - [“About the Db2 Cloning Tool stored procedure” on page 153](#)
 - [“Migrating the stored procedure from Db2 Cloning Tool V3.1” on page 154](#)
 - [“Calling the stored procedure” on page 167](#)
 - [“Parameter files and parameter descriptions” on page 155](#)
 - [“Stored procedure example: Cloning a non-data sharing subsystem” on page 172](#)
 - [“Recloning a Db2 system” on page 171](#)
 - [“Building and running the verified cloning” on page 170](#)
 - [“Verifying the cloning” on page 169](#)
- Updates throughout the subsystem cloning portions of the user guide were made for the Db2 directory page name change from "DBD01" for Db2 10 CM to "SYSDBDXA" for Db2 10 NFM and later.
- **Db2 Cloning Tool Table Space Cloning**
 - The EXCLUDE-MISMATCH-PROCESSING command was added to address storage abends that might occur during mismatch checking while processing a large number of data sets in the source job. The topics [“COPY command syntax” on page 514](#) and [“COPY command and keyword definitions” on page 516](#) were updated and message CKZ50606I was added to support this feature.
 - The topic [Chapter 20, “Options for rebuilding indexes,” on page 297](#) was added to better describe options for creating and submitting the JCL to rebuild indexes when cloning table spaces using image copies as the source, or use the log apply or data masking features.
 - A clarification was added to the topic [Chapter 18, “Using image copies to clone table spaces and index spaces,” on page 289](#) to explain how Db2 Cloning Tool Table Space Cloning checks for mismatches when cloning using source image copies.
 - The topics [“Function authorization requirements” on page 23](#) and [“TCP/IP server job details \(optional\)” on page 211](#) were updated to describe the authorizations required when using the TCP/IP server job as a started task.
 - The RI parameter of the LISTDEF command was modified to clarify how it works with INCLUDE and EXCLUDE parameters. See the topic [“LISTDEF command and keyword definitions” on page 565](#).
 - The table space cloning LISTDEF command now has a defined default of ALL. The topic [“LISTDEF command and keyword definitions” on page 565](#) was updated.
 - Information was added to the data masking MASKRULE (STATIC) topic to describe how a VARCHAR field is masked when the VARCHAR field value is shorter than the provided mask value. The topic [“MASKRULE\(STATIC, numeric | “date_or_time” | “string”\)” on page 273](#) was updated.
 - The ALTER-FOR-XML-LOB-COLUMNS parameter was added, which can be used for DDL processing to avoid column mismatches when the source table space has XML or LOB columns and has been altered since the XML or LOB columns were defined. The topics [“COPY command syntax” on page 514](#) and [“COPY command and keyword definitions” on page 516](#) were updated and a message was added to support this feature. The ISPF interface topic [COPY command defaults](#) also was updated.
 - The SUPPRESS-RI-CONSTRAINTS keyword was added to provide with the option to not create referential constraints in generated DDL. The topics [“COPY command syntax” on page 514](#) and [“COPY command and keyword definitions” on page 516](#) were updated. The ISPF interface topic [COPY command defaults](#) also was updated.
- **Changes to the ISPF interface**
 - The ISPF interface now requires a minimum available region size of 20000 KB (reduced from 30000 KB). The topics [“Set up your environment prior to customization” on page 21](#) and [The Db2 Cloning Tool ISPF interface](#) were updated, and message CKZ002E was updated.

- Db2 Cloning Tool now supports the DISPLAY MEPL command on the ISPF interface. This command can be used to provide maintenance level information and diagnostic information to assist customer support. The topic [“Gathering maintenance level information with the DISPLAY MEPL command” on page 1081](#) was added, and messages were added to support this feature.
- The RESET command was added to the user defaults panels to allow you to reset all default values for subsystem and table space cloning commands to installation defaults. The topics [Setting subsystem cloning defaults](#) and [Setting table space cloning defaults](#) were updated.
- Subsystem settings were removed from several table space cloning panels. You must set subsystem settings using the **Source and Target DB2 subsystem** panel. See the topic [“Select the source and target Db2 subsystems” on page 350](#).
- The WARN-ON-VERSIONS command was added to [COPY command defaults](#).
- The UTILITY-COMMAND-EXECUTE-PERCENT command was added to [SET command defaults](#).
- Incorrect values for the RENAME-AUDIT-LOG and RENAME-AUDIT-LOG SMF fields were corrected in [RENAME command defaults](#).
- Text was clarified in several locations to state that subsystem and table space cloning defaults are saved in a VSAM profile repository and are specific to each TSO user ID and LPAR.

Product legal notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive

Armonk, NY 10504-1785
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.html>.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

A

accessibility
 overview [16](#)

ACF authorities [109](#)

ACTIVELOG-LIMIT
 DB2RBLDBSDS [433](#)

ADVISORY-STATUS-VALUES, table space cloning [568](#)

ALLOW-COPY-ON-MISMATCH, table space cloning [517](#)

ALLOW-SKIP-INACTIVE-MEMBERS, table space cloning [531](#)

ALWAYS-COPY-HISTORY-TABLES, table space cloning [517](#)

ALWAYS-COPY-INDEXSPACES, table space cloning [518](#)

APF authorization [30](#)

application cloning
 building jobs [319](#)
 from image copies [316](#)
 from source Db2 data sets [315](#)
 submitting jobs for application cloning from user copies [321](#)
 submitting jobs for application cloning without shared DASD [322](#)
 submitting jobs for cloning from Db2 data sets [321](#)
 submitting jobs for cloning from image copies [321](#)
 using user copies [316](#)
 verifying DD specifications for the cloning jobs [320](#)
 without shared DASD [317](#)

application cloning profiles
 creating [313](#), [314](#)
 exporting and importing [99–101](#)
 managing [315](#)
 selecting source objects [318](#), [319](#)
 selecting the source and target Db2 SSIDs [318](#)
 working with [313](#)

application cloning profiles, migrating from table space cloning profiles [103](#)

authorization requirements [23](#)

authorizations [97](#)

AUTO_START_SOURCE_SPACE keyword [1146](#)

AUTO_START_TARGET_SPACE keyword [1146](#)

AUTO_STOP_TARGET_SPACE keyword [1146](#)

AUTO-INDEXSPACE-TRANSLATE, table space cloning [518](#)

AUTO-START-SOURCE-SPACE, table space cloning [519](#)

AUTO-START-TARGET-SPACE, table space cloning [519](#)

AUTO-STOP-TARGET-SPACE, table space cloning [519](#)

AUTO-TABLESPACE-TRANSLATE, table space cloning [520](#)

B

BACKINFO-DDN
 DB2GETBACKINFO [426](#)
 RESTORE-FROM-DUMPTAPES command [491](#)

BACKINFO-REFORMAT command
 Command and keyword definitions [374](#)
 JCL [376](#)

BACKINFO-REFORMAT command syntax [373](#)

BACKUP
 UCATOPTIONS command [497](#)

BACKUP FORCE
 UCATOPTIONS command [497](#)

BASE-JOURNAL-DDN
 COPY command [396](#)

BCSCLEAN
 target catalog available to applications [110](#)

BCSCLEAN command
 CLEANUP-CATALOG-DSNMASKS [380](#)
 CLEANUP-CATALOG-ORPHANS [380](#)
 Command and keyword definitions [380](#)
 Delete previous application cycle entries from target catalog [379](#)
 JCL [382](#)
 SIMULATE [381](#)

BCSCLEAN command syntax [379](#)

BSDS-DDN
 DB2GETBACKINFO [426](#)

C

Capture catalog data
 COPY step JCL [402](#)
 COPY-BY-DS step JCL [411](#)

Catalog names
 About source and target [383](#)

CATALOG-PREFETCH, table space cloning [520](#)

catalogs
 populated or empty [110](#)

CATWORK_ATTR keyword [1146](#)

CATWORK-ATTR
 COPY command [396](#)

CATWORK-DSN
 COPY command [388](#)

CHECK-DATASET-COMPATIBILITY, table space cloning [522](#)

CHECK-INDEX-KEYS, table space cloning [522](#)

CKZ
 JCLFINDUCATS [469](#)

CKZINI member
 customization values [1144](#)
 structure [1145](#)
 syntax rules [1145](#)
 token errors [1146](#)

CKZINI member keywords
 AUTO_START_SOURCE_SPACE [1146](#)
 AUTO_START_TARGET_SPACE [1146](#)
 AUTO_STOP_TARGET_SPACE [1146](#)
 CATWORK_ATTR [1146](#)
 CONCURRENT_EXECUTIONS [1146](#)
 CONCURRENT_EXECUTIONS_WAIT_TIME [1146](#)
 COPY_IF_NO_DB2_TARGET_OBJECTS [1146](#)
 DB2_CLONING_TOOL_REL_DATE [1146](#)
 DB2_CLONING_TOOL_REL [1146](#)
 DB2_PLAN [1146](#)
 DB2_XCFCLEAN [1146](#)
 DB2V11_EPIC37738_INSTALLED [1146](#)
 DSNS_PER_COPY [1146](#)
 DSS_COPY_COMMANDS [1146](#)

CKZINI member keywords (*continued*)

- [GDG_ALL_MIGRATED 1146](#)
- [GDG_ALL_MIGRATED_RETAIN_RC 1146](#)
- [GDG_EMPTY 1146](#)
- [GDG_EMPTY_RETAIN_RC 1146](#)
- [GDG_MIGRATED 1146](#)
- [GDG_MIGRATED_RETAIN_RC 1146](#)
- [GDG_TAPE 1146](#)
- [GDG_TAPE_RETAIN_RC 1146](#)
- [INIMERGE_VALUES 1146](#)
- [ISSUE_CKZ14141I 1146](#)
- [ISSUE_VCLOSE 1146](#)
- [ISSUE_VCLOSE_SCOPE 1146](#)
- [MAX_COPY_RC 1146](#)
- [MAX_RC 1146](#)
- [MAX_RENAME_TASKS 1146](#)
- [MAX_SUBTASKS 1146](#)
- [MIM_GDIF 1146](#)
- [MISSING_USERCAT_DISP 1146](#)
- [MISSING_USERCAT_RC 1146](#)
- [NOT_RENAMED_DISP 1146](#)
- [NOT_RENAMED_RC 1146](#)
- [ORPHAN_CATENTRY_DISP 1146](#)
- [ORPHAN_CATENTRY_RC 1146](#)
- [RECATALOG 1146](#)
- [RENAME_ERROR 1146](#)
- [RENAME_ERROR_CONTINUE_RC 1146](#)
- [RESET_LOGRBA 1146](#)
- [SPACE_MANAGEMENT 1146](#)
- [SPECIAL_SECTIONS 1146](#)
- [TARGET_VOLS_SHOULD_BE_EMPTY 1146](#)
- [TCPIP_SERVER_PORT 1146](#)
- [TCPIP_STC_NAME 1146](#)
- [TEMP_DATASET_DISP 1146](#)
- [TEMP_DATASET_RC 1146](#)
- [VTOCIX_REBUILDER 1146](#)

classic ISPF interface [103, 323](#)

CLEANUP-CATALOG-DSNMASKS

- [BCSCLEAN command 380](#)

CLEANUP-CATALOG-ORPHANS

- [BCSCLEAN command 380](#)

CLONED, table space cloning [566](#)

cloning

- Db2 subsystem [107](#)
- subsystem [1105, 1115, 1128, 1134, 1139](#)
- subsystem by using data set copy [1103](#)
- volumes [1091, 1096](#)
- with DB2 Administration Tool [369](#)

cloning Db2 subsystem [112](#)

cloning definition [13](#)

cloning from image copies

- steps [290](#)

cloning from image copies source job changes [292](#)

cloning scenarios [1091](#)

cloning source VSAM files using Sterling Connect:Direct [310](#)

cluster names, VSAM [107](#)

CMDDDDNAME, table space cloning [525](#)

COLD-START

- [DB2ALTERBSDS 416](#)

Command and keyword definitions

- [BACKINFO-REFORMAT command 374](#)
- [BCSCLEAN command 380](#)
- [COPY command 388](#)
- [COPY command, table space cloning 516](#)

Command and keyword definitions (*continued*)

- [COPY-BY-DS command 407](#)
- [COPYCHECK command 413](#)
- [DATASUBTYPE command 560](#)
- [DB2ALTERBSDS command 416](#)
- [DB2FIX command 422](#)
- [DB2GETBACKINFO command 426](#)
- [DB2LGRNXCLEAN command 430](#)
- [DB2RBLDBSDS command 433](#)
- [DB2SCHEMA-UPDATE command 437](#)
- [DB2SETLOG command 440](#)
- [DB2SQL command 444](#)
- [DB2START command 449](#)
- [DB2STOP command 453](#)
- [DB2UPDATE command 456](#)
- [DB2UTILXCLEAN command 464](#)
- [DB2XCFCLEAN command 467](#)
- [FINDUCATS command 469](#)
- [HLQDDDF command 561](#)
- [JRNLUPGRADE command 471](#)
- [LISTDEF command 565](#)
- [ONLINECLIP command 473](#)
- [RENAME command 477](#)
- [RESTORE-FROM-DUMPTAPES command 491](#)
- [SET command 568](#)
- [UCATOPTIONS command 497](#)
- [VARYOFF command 500](#)
- [VARYON command 504](#)
- [VOLOPTIONS command 508](#)

commands [513](#)

Commands

- [BACKINFO-REFORMAT 372](#)
- [BCSCLEAN 379](#)
- [COPY 383, 513](#)
- [COPY-BY-DS 404](#)
- [COPYCHECK 412](#)
- [DATASUBTYPE 560](#)
- [DB2ALTERBSDS 414](#)
- [DB2FIX 420](#)
- [DB2GETBACKINFO 424](#)
- [DB2LGRNXCLEAN 430](#)
- [DB2RBLDBSDS 432](#)
- [DB2SCHEMA-UPDATE 435](#)
- [DB2SETLOG 440](#)
- [DB2SQL 442](#)
- [DB2START 447](#)
- [DB2STOP 452](#)
- [DB2UPDATE 454](#)
- [DB2UTILXCLEAN 463](#)
- [DB2XCFCLEAN 466](#)
- [FINDUCATS 468](#)
- [HLQDDDF 561](#)
- [JRNLUPGRADE 470](#)
- [LISTDEF 562](#)
- [ONLINECLIP 472](#)
- [RENAME 474](#)
- [RESTORE-FROM-DUMPTAPES 489](#)
- [SET 567](#)
- [UCATOPTIONS 496](#)
- [VARYOFF 499](#)
- [VARYON 503](#)
- [VOLOPTIONS 507](#)

COMMIT-FREQUENCY, table space cloning [544](#)

COMMON-CONSISTENT-POINT, table space cloning [531](#)

CONCURRENT_EXECUTIONS keyword [1146](#)
 CONCURRENT_EXECUTIONS_WAIT_TIME keyword [1146](#)
 configuration tasks
 Db2 version migration considerations [105](#)
 configuring Db2 subsystems for ISPF interface (classic) [324](#)
 configuring Db2 subsystems for ISPF interface(new) [312](#)
 CONNECT-DB2-ON-CLIENT-CONNECT, table space cloning [569](#)
 cookie policy [1183](#), [1185](#)
 copies, volumes
 planning for [107](#)
 COPY
 Initiate volume copies and catalog data capture [402](#)
 Step JCL [402](#)
 copy by volume [115](#)
 COPY command
 BASE-JOURNAL-DDN [396](#)
 CATWORK-ATTR [396](#)
 CATWORK-DSN [388](#)
 Command and keyword definitions [388](#)
 DATA-MOVER [397](#), [401](#)
 DATA-MOVER(PGM(ADRSSU)) [401](#)
 EXCLUDE-FROM-VOLSER [399](#)
 EXCLUDE-TO-VOLSER [399](#)
 FROM-STORAGEGROUP [391](#)
 FROM-USER-STORAGEGROUP [391](#)
 FROM-VOLSER [392](#)
 FROM-VOLSER-DDN [392](#)
 JCL [402](#)
 JOURNAL-DDN [388](#)
 JOURNAL-DSN [388](#)
 Keyword definitions [388](#)
 NOUSERCATALOGS [391](#)
 SIMULATE [400](#)
 Source and target names [383](#)
 SOURCEONLINE [400](#)
 Step JCL [402](#)
 TARGET-UCATS-ON-TARGET-VOLUMES [401](#)
 TARGET-VOLS-SHOULD-BE-EMPTY [401](#)
 TARGETSONLINE [401](#)
 TARGETSUONLINE [401](#)
 TO-STORAGEGROUP [392](#)
 TO-USER-STORAGEGROUP [393](#)
 TO-VOLSER [393](#), [394](#)
 USERCATALOGS [388](#)
 USERCATALOGS-BACKUPFIRST [390](#)
 USERCATALOGS-NOBACKUP [389](#)
 USERSGDEFS-DDN [394](#)
 USERSGDEFS-OFFSETS [394](#)
 VOLPAIRS [395](#)
 VOLPAIRS-DDN [395](#)
 VOLPAIRSDEVN [395](#)
 VOLPAIRSDEVN-DDN [395](#)
 VOLPAIRSDEVN-NOCLIP [395](#)
 VOLPAIRSDEVN-NOCLIP-DDN [396](#)
 COPY command syntax [386](#)
 COPY command syntax, table space cloning [514](#)
 COPY command, table space cloning [516](#)
 Copy DB2 Entries panel [92](#)
 COPY step
 Sample control statements [402](#)
 COPY_IF_NO_DB2_TARGET_OBJECTS keyword [1146](#)
 COPY-BY-DS
 Initiate volume copies and catalog data capture [411](#)
 COPY-BY-DS (*continued*)
 Step JCL [411](#)
 COPY-BY-DS command
 Command and keyword definitions [407](#)
 DATA-MOVER [409](#)
 EXCLUDE-SRCNAME-MASKS [409](#)
 EXCLUDE-SRCNAME-MASKS-DDN [410](#)
 JCL [411](#)
 JOURNAL-DDN [408](#)
 JOURNAL-DSN [408](#)
 Keyword definitions [407](#)
 MIGRATED-DSN [410](#)
 OFFLINE [410](#)
 RENAME-MASKS [408](#)
 RENAME-MASKS-DDN [408](#)
 REPLACE-UNCONDITIONAL [411](#)
 SIMULATE [411](#)
 Step JCL [411](#)
 COPY-BY-DS command syntax [407](#)
 COPY-BY-DS step
 Sample control statements [411](#)
 COPY-CMD-DDN
 VARYOFF command [500](#)
 COPY-IF-NO-DB2-TARGET-OBJECTS, table space cloning [522](#)
 COPY-IJ-TO-NONEXISTENT-TARGET, table space cloning [523](#)
 COPYCHECK
 Use to scan status of volume copies [412](#)
 Use to withdraw previously established volume relationships [412](#)
 COPYCHECK command
 Command and keyword definitions [413](#)
 JCL [414](#)
 JOURNAL-DDN [413](#)
 JOURNAL-DSN [413](#)
 relationships conflicts [109](#)
 STOPSNAP [414](#)
 WAIT [413](#)
 WITHDRAW [413](#)
 COPYCHECK command syntax [412](#)
 coupling facility structures, deallocating [1160](#)
 create volume copies [115](#), [116](#)
 customization
 after customizing using Tools Customizer [97](#), [98](#)
 associated list
 adding Db2 entries [83](#)
 overview [1085](#)
 associating Db2 entries [83](#)
 browsing parameters [92](#)
 changing parameters [77](#)
 component [1085](#)
 copying Db2 entries [92](#)
 Create a DB2 Entry panel [83](#)
 creating Db2 entries [83](#)
 customization jobs
 deleting [95](#)
 displaying [95](#)
 generating [90](#)
 maintaining [95](#)
 regenerating [90](#)
 renaming [95](#)
 sort sequence [90](#)
 submitting [90](#), [95](#)

customization (*continued*)

- customization library
 - deleting jobs [95](#)
 - maintaining [95](#)
 - overview [1087](#)
 - recustomizing [95](#)
 - renaming jobs [95](#)
- customization library qualifier
 - specifying [72](#)
- Customized status [1085](#)
- Customizer Workplace panel [90](#)
- customizing a new version of a product [77](#)
- customizing a product for the first time [77](#)
- customizing settings [72](#)
- data sets
 - customization library [1087](#)
 - data store [1087](#)
 - Discover EXEC library [1087](#)
 - metadata library [1087](#)
- data store
 - overview [1087](#)
- data store data set
 - specifying [72](#)
- Db2 data sharing members
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
- Db2 entries
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
 - defining [90](#)
 - deleting [94](#)
 - generating jobs for [90](#)
 - removing [94](#)
 - selecting [90](#)
 - specifying [90](#)
 - unassociating [94](#)
- Db2 group attach field
 - specifying [72](#)
- Db2 group attach names
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
- Db2 parameters
 - defining [88](#)
 - editing [88](#)
- DB2 Parameters panel [88](#)
- Db2 subsystems
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
- defining Db2 parameters [88](#)
- defining LPAR parameters [87](#)
- defining parameters [85](#), [90](#)
- defining product parameters [85](#)
- deleting Db2 entries [94](#)
- deleting jobs [79](#)
- Discover Customized Product Information panel [81](#)
- Discover EXEC

customization (*continued*)

- Discover EXEC (*continued*)
 - customizing a new version of a product [77](#), [78](#)
 - overview [1087](#)
 - retrieving product information automatically [81](#)
- Discovered status [1085](#)
- discovering previous versions [78](#)
- discovering product information [81](#)
- displaying jobs [95](#)
- editing LPAR parameters [87](#)
- editing parameters [77](#)
- editing product parameters [85](#)
- Errors in Customization status [1085](#)
- finding trace data set [1084](#)
- Finish Product Customization panel [90](#)
- first-time [77](#)
- first-time customization [77](#)
- generating jobs [90](#)
- high-level qualifier [1085](#)
- Incomplete status [1085](#)
- job sort order [90](#)
- jobs
 - deleting [95](#)
 - displaying [95](#)
 - maintaining [95](#)
 - renaming [95](#)
 - sort order [90](#)
 - submitting [90](#), [95](#)
- LPAR parameters
 - defining [87](#)
 - editing [87](#)
- LPAR Parameters panel [87](#)
- LPARs [96](#)
- maintaining jobs [95](#)
- master list
 - adding Db2 entries [83](#)
 - Associate DB2 Entry for Product panel [83](#)
 - overview [1085](#)
- metadata libraries
 - specifying [80](#)
- metadata library
 - overview [1087](#)
 - specifying [72](#)
- modifying parameters [77](#)
- modifying settings [72](#)
- multiple instances [72](#)
- multiple-LPAR environment [96](#)
- Not Required status [1085](#)
- panels
 - Associate DB2 Entry for Product [83](#)
 - Create a DB2 Entry [83](#)
 - Customizer Workplace [90](#)
 - DB2 Parameters [88](#)
 - Discover Customized Product Information [81](#)
 - Finish Product Customization [90](#)
 - LPAR Parameters [87](#)
 - Product Parameters [85](#)
 - Specify the Metadata Library [80](#)
- parameter values [31](#)
- parameters
 - browsing [92](#)
 - defining [85](#), [90](#)
 - viewing [92](#)
- preparing to use Tools Customizer [72](#)

- customization (*continued*)
 - product [1085](#)
 - product parameters
 - changing [79](#)
 - defining [85](#)
 - editing [79, 85](#)
 - modifying [79](#)
 - Product Parameters panel [85](#)
 - Ready to Customize status [1085](#)
 - recustomization [77, 79](#)
 - recustomizing [79](#)
 - recustomizing a product [77](#)
 - removing Db2 entries [94](#)
 - roadmaps
 - customizing for the first time [77](#)
 - first-time customization [77](#)
 - reustomizing [79](#)
 - Specify the Metadata Library panel [80](#)
 - specifying data sets [72](#)
 - specifying metadata libraries [80](#)
 - starting Tools Customizer [71](#)
 - status types
 - Customized [1085](#)
 - Discovered [1085](#)
 - Errors in Customization [1085](#)
 - Incomplete [1085](#)
 - Not Required [1085](#)
 - Ready to Customize [1085](#)
 - submitting jobs [90](#)
 - terminology [1085](#)
 - trace data set [1084](#)
 - troubleshooting
 - finding trace data set [1084](#)
 - user job card settings
 - specifying [72](#)
 - viewing parameters [92](#)
- customization library
 - overview [1087](#)
- customization library qualifier
 - specifying [72](#)
- customizing settings [72](#)

D

- data masking
 - rebuilding indexes [281](#)
 - sample job [282](#)
 - specifying masking rules [273](#)
 - summary of steps [271](#)
 - target job changes [281](#)
- data masking source job changes [272](#)
- data set names
 - gathering [27](#)
- data set renaming considerations
 - application name length restriction [109](#)
- data set/sphere integrity
 - GDGs [107](#)
 - multivolume data sets [107](#)
 - non-VSAM data set aliases [107](#)
 - VSAM spheres [107](#)
- data sets
 - multivolume [107](#)
 - renamed and not named [107](#)
- data store

- data store (*continued*)
 - overview [1087](#)
- data store data set
 - specifying [72](#)
- DATA-MASKING, table space cloning [523](#)
- DATA-MOVER
 - COPY command [397, 401](#)
 - COPY-BY-DS command [409](#)
 - DATA-MOVER PGM(ADDRSSU), table space cloning [524](#)
 - DATA-MOVER PGM(EMCAPI), table space cloning [524](#)
 - DATA-MOVER PGM(NONE), table space cloning [525](#)
 - DATA-MOVER PGM(SRCIMCPY), table space cloning [524](#)
 - DATA-MOVER PGM(SRCVSCPY), table space cloning [524](#)
 - DATA-MOVER, table space cloning [524](#)
 - DATA-MOVER(PGM(ADDRSSU))
 - Discrete VOLSER specification [401](#)
 - Source volumes [401](#)
 - Storage group specification [402](#)
 - Target volumes [402](#)
 - VOLSER mask specification [401](#)
- DATA-SHARING-MEMBERS, table space cloning [531](#)
- DATABASES
 - DB2FIX [422](#)
- DATABASESONLY
 - DB2GETBACKINFO [427](#)
- DATACLAS
 - RENAME command [478](#)
- DATACLAS-PAIRS
 - RENAME command [478](#)
- DATACLAS, table space cloning [525](#)
- DATASETS-TO-COPY-DDN, table space cloning [526](#)
- DATASUBTYPE [560](#)
- DATASUBTYPE command syntax [560](#)
- DATASUBTYPE, table space cloning [560](#)
- Db2 [10 200](#)
- Db2 12 function level support [9](#)
- DB2 Administration Tool, using DB2 Cloning Tool with [369](#)
- DB2 Administration Tool for z/OS, adding to Launchpad [99](#)
- Db2 BACKUP SYSTEM [1115](#)
- Db2 BACKUP SYSTEM command [1105, 1128](#)
- Db2 Cloning Tool
 - cloning definition [13](#)
 - features and benefits [15](#)
 - JCL DB2ALTERBSDS [419](#)
 - JCL DB2SETLOG [442](#)
 - JCL DB2UPDATE [461, 462](#)
 - job templates [303](#)
 - overview [12](#)
 - reasons for using [13](#)
 - support [13](#)
 - why clone a subsystem? [13](#)
- Db2 Cloning Tool Table Space Cloning
 - checking object attribute compatibility [192](#)
 - CKZFIX job [216](#)
 - considerations for in-progress read/write activity or Db2 utilities [183](#)
 - copy procedure [261–270](#)
 - copy procedures [261](#)
 - Db2 entity terminology [181](#)
 - Db2 Version 9.1 and later considerations [200](#)
 - description of [179](#)
 - features and benefits [180](#)
 - identity columns [184](#)
 - implicitly created objects [184](#)

- Db2 Cloning Tool Table Space Cloning *(continued)*
 - index caching [202](#)
 - object mismatch processing [191](#), [194](#)
 - overview [179](#)
 - planning for catalog prefetch [201](#)
 - planning for cloning [183](#)
 - renaming table spaces and index spaces [183](#)
 - report job (optional) [213](#)
 - rules for using [183](#)
 - runtime repository functionality [216](#)
 - set up for other methods [237](#)
 - set up procedures for FlashCopy, SnapShot, or TimeFinder/Clone [219](#)
 - setup for copy by data set [219](#), [226](#), [237](#), [246](#)
 - setup for FlashCopy, SnapShot, or TimeFinder/Clone [219](#), [226](#)
 - setup for other copy methods [246](#)
 - source job [208](#)
 - source job - copy phase IV [210](#)
 - source job - discovery phase II [208](#)
 - source job - eligibility phase III [209](#)
 - source job - init phase I [208](#)
 - steps for copying by data set using other methodologies [241–243](#), [248](#), [252–254](#), [262](#)
 - steps for copying by data set with FlashCopy, SnapShot, or TimeFinder/Clone [219](#), [220](#), [226](#), [227](#)
 - steps for copying by data set with FlashCopy/SnapShot [223](#), [231–233](#), [238](#), [244](#), [252](#), [255](#), [256](#)
 - steps for copying by data set with other copy methodologies [247](#), [256](#)
 - steps for copying by data set with other methodologies [237](#)
 - strategies for resolving object mismatches [194](#)
 - subsystem copy rules [183](#)
 - target job [211](#)
 - TCP/IP server job (optional) [211](#)
 - why you should use [180](#)
- Db2 group attach field
 - specifying [72](#)
- Db2 Recovery Expert backup
 - cloning when backup volumes are online [1134](#)
- Db2 Recovery Expert for z/OS backup
 - cloning when backup volumes are offline [1139](#)
- Db2 subsystem cloning
 - cloning to a specific point in time using a BACKUP SYSTEM backup [1115](#)
- Db2 Version 9.1 [200](#)
- Db2 version migration considerations [105](#)
- DB2_CLONING_TOOL keywords [1146](#)
- DB2_CLONING_TOOL_REL keywords [1146](#)
- DB2_PLAN keyword [1146](#)
- DB2_XFCLEAN keyword [1146](#)
- DB2-COMMAND-RESPONSE-WAIT, table space cloning [569](#)
- DB2-MEMBER
 - DB2ALTERBSDS [416](#)
- DB2-NAME
 - DB2LGRNXCLEAN [431](#)
- DB2-PLAN, table space cloning [570](#)
- DB2-SSID
 - DB2FIX [422](#)
- DB2ALTERBSDS [414](#)
- DB2ALTERBSDS command
 - COLD-START [416](#)
 - Command and keyword definitions [416](#)

- DB2ALTERBSDS command *(continued)*
 - DB2-MEMBER [416](#)
 - JCL [418](#)
 - JOURNAL-DDN [416](#)
 - JOURNAL-DSN [416](#)
 - Step JCL [419](#)
- DB2ALTERBSDS command syntax [415](#)
- DB2FIX command
 - Command and keyword definitions [422](#)
 - DATABASES [422](#)
 - DB2-SSID [422](#)
 - DSNDB01-DBD01-STARTED [422](#)
 - EXCLUDE-MASKS [422](#)
 - JCL [424](#)
 - MAX-CONCURRENT-CMDS [422](#)
 - MEMBERS-AND-DBD01 [423](#)
 - MEMBERS-NEED-STARTING [423](#)
 - SIMULATE [423](#)
 - START-SCOPE [423](#)
 - WAIT [423](#)
 - WAIT-AND-DBD01 [424](#)
- DB2FIX command syntax [421](#)
- DB2GETBACKINFO command
 - BACKINFO-DDN [426](#)
 - BSDS-DDN [426](#)
 - Command and keyword definitions [426](#)
 - DATABASESONLY [427](#)
 - DUMP-CLASS [427](#)
 - JCL [428](#)
 - LAST [427](#)
 - LIST [427](#)
 - LOCATION [426](#)
 - LOGSONLY [427](#)
 - TOKEN [427](#)
 - USE-DUMPTAPES [428](#)
 - USERCATALOGS [428](#)
 - WORK-DDN [426](#)
 - WORK-DSN [426](#)
- DB2GETBACKINFO command syntax [425](#)
- DB2LGRNXCLEAN
 - JOURNAL-DSN [431](#)
- DB2LGRNXCLEAN command
 - Command and keyword definitions [430](#)
 - DB2-NAME [431](#)
 - JCL [431](#)
- DB2LGRNXCLEAN command syntax [430](#)
- DB2RBLDBSDS
 - JOURNAL-DSN [433](#)
- DB2RBLDBSDS command
 - ACTIVELOG-LIMIT [433](#)
 - Command and keyword definitions [433](#)
 - JCL [434](#)
- DB2RBLDBSDS command syntax [432](#)
- DB2SCHEMA-UPDATE command
 - Command and keyword definitions [437](#)
 - JCL [439](#)
- DB2SCHEMA-UPDATE command syntax [436](#)
- DB2SETLOG command
 - Command and keyword definitions [440](#)
 - JCL [441](#)
 - Step JCL [442](#)
- DB2SETLOG command syntax [440](#)
- DB2SQL command
 - Command and keyword definitions [444](#)

DB2SQL command (*continued*)
 JCL [447](#)

DB2START command
 Command and keyword definitions [449](#)
 JCL [451](#)

DB2START command syntax [449](#)

DB2STOP command
 Command and keyword definitions [453](#)
 JCL [454](#)

DB2STOP command syntax [452](#)

DB2UPDATE command
 Command and keyword definitions [456](#)
 JCL [460](#)
 Step JCL [461](#), [462](#)

DB2UPDATE command syntax [455](#)

DB2UTILXCLEAN command
 Command and keyword definitions [464](#)
 JCL [465](#)
 JCL for identifying objects with UT status [465](#)

DB2UTILXCLEAN command syntax [464](#)

DB2V11_EPIC37738_INSTALLED keyword [1146](#)

DB2XCFCLEAN command
 Command and keyword definitions [467](#)
 JCL [468](#)

DB2XCFCLEAN command syntax [466](#)

DBSQL command syntax [443](#)

DDL-ATTRIBUTE-CHANGE, table space cloning [544](#)

DDL-ENABLE, table space cloning [543](#)

DDL, processing [195](#)

DDL, re-creating target objects [196](#)

DDL, scenarios for processing [198](#)

deallocating coupling facility structures [1160](#)

dedicated volumes
 conserve space [107](#)

DEFAULT_IF_NO_MATCH
 RENAME command [478](#), [487](#)

DEFAULT-SQLID, table space cloning [570](#)

Delete previous application cycle entries using BCSCLEAN [379](#)

DELETE request
 for table space cloning via REST API [366](#)

DELETE-MINILOG-DATASETS, table space cloning [531](#)

deprecated functions and parameters [10](#)

DFSMS data encryption [111](#)

diagnostic information
 gathering [1080](#), [1081](#), [1083](#)

DIRECTION
 HLQDDDF command [562](#)

Discover EXEC
 overview [1087](#)

discrete vols vs.SMS storage group specs [107](#)

Discrete VOLSER specification
 DATA-MOVER(PGM(ADRSSU)) [401](#)

DISPLAY MEPL command [1081](#)

documentation
 accessing [16](#)
 sending feedback [16](#)

documentation changes [1](#), [1161](#)

DRIVEACS
 RENAME command [478](#)

DSNDB01-DBD01-STARTED
 DB2FIX [422](#)

DSNS_PER_COPY keyword [1146](#)

DSNS-PER-COPY, table space cloning [527](#)

DSS_COPY_COMMANDS keyword [1146](#)

DSS-COPY-COMMANDS, table space cloning [527](#)

dump tapes, using for cloning [1128](#)

DUMP-CLASS
 DB2GETBACKINFO [427](#)

dynamic rebuild of indexes [297](#)

E

EBCDIC code set [25](#), [181](#)

EMC TimeFinder/Clone Mainframe Snap Facility data set level support [179](#)

EMC TimeFinder/Mirror target volume clones [122](#)

EMC TimeFinder/Mirror Target Volume Clones [122](#)

EMCAPI
 cloning table spaces with [558](#)
 GLOBAL command parameter values for [558](#)

empty ICF catalog [110](#)

ENABLE-PREFETCH, table space cloning [520](#)

ENABLE-SOURCE-PREFETCH, table space cloning [521](#)

ENABLE-TARGET-PREFETCH, table space cloning [521](#)

ENCRYPTION-MISMATCH-RC, table space cloning [527](#)

encryption, pervasive [111](#)

END-POINT, table space cloning [532](#)

EXCLUDE-FROM-VOLSER
 COPY command [399](#)
 FINDUCATS command [469](#)

EXCLUDE-MASKS
 DB2FIX [422](#)

EXCLUDE-MISMATCH-PROCESSING, table space cloning [528](#)

EXCLUDE-SRCNAME
 RENAME command [479](#)

EXCLUDE-SRCNAME-MASKS
 COPY-BY-DS command [409](#)
 RENAME command [479](#)

EXCLUDE-SRCNAME-MASKS-DDN
 COPY-BY-DS command [410](#)
 RENAME command [479](#)

EXCLUDE-TO-VOLSER
 COPY command [399](#)
 RESTORE-FROM-DUMPTAPES command [492](#)

EXCLUDEFROMVOLSER
 FINDUCATS command [469](#)

EXPLODE-OBJECTS, table space cloning [544](#)

EXPORT command
 for exporting ISPF profiles and control files [100](#)

F

FASTREP, table space cloning [525](#)

FCTOPPRCPRIAMRY, table space cloning [525](#)

FCTOXRCPRIAMRY, table space cloning [526](#)

Filter masks
 Newname syntax [476](#)
 Oldname syntax [476](#)
 Oldvalue syntax [374](#)
 Rename considerations [475](#)

filter pattern masks [26](#)

FINDUCATS [468](#)

FINDUCATS command
 Command and keyword definitions [469](#)

EXCLUDE-FROM-VOLSER [469](#)

FINDUCATS command (*continued*)
 EXCLUDEFROMVOLSER [469](#)
 FROM-STORAGEGROUP [469](#)
 FROM-VOLSER [469](#)
 FROMSTORAGEGROUP [469](#)
 FROMVOLSER [469](#)
 JCL [469](#)
 Keyword definitions [469](#)
 Step JCL [469](#)
 FINDUCATS command syntax [468](#)
 first-time customization [77](#)
 Flash Copy
 source and target volume pairing [107](#)
 FlashCopy [115](#), [179](#)
 FROM-STORAGEGROUP
 COPY command [391](#)
 FINDUCATS command [469](#)
 FROM-USER-STORAGEGROUP
 COPY command [391](#)
 FROM-VOLSER
 COPY command [392](#)
 FINDUCATS command [469](#)
 FROM-VOLSER-DDN
 COPY command [392](#)
 FROMSTORAGEGROUP
 FINDUCATS command [469](#)
 FROMVOLSER
 FINDUCATS command [469](#)
 function authorization requirements [23](#)
 FUZZY-COPY, table space cloning [528](#)

G

GDG_ALL_MIGRATED keyword [1146](#)
 GDG_ALL_MIGRATED_RETAIN_RC keyword [1146](#)
 GDG_EMPTY keyword [1146](#)
 GDG_EMPTY_RETAIN_RC keyword [1146](#)
 GDG_MIGRATED keyword [1146](#)
 GDG_MIGRATED_RETAIN_RC keyword [1146](#)
 GDG_TAPE keyword [1146](#)
 GDG_TAPE_RETAIN_RC keyword [1146](#)
 GDG-ALL-MIGRATED
 RENAME command [480](#)
 GDG-EMPTY
 RENAME command [480](#)
 GDG-MIGRATED
 RENAME command [480](#)
 GDG-TAPE
 RENAME command [480](#)
 GDGs
 base names matching rename mask [107](#)
 source volume list [107](#)
 GENERATE-DDL-DEFAULTS, table space cloning [544](#)
 GMT-OFFSET, table space cloning [532](#)

H

HISTORY, table space cloning [565](#)
 HLQDDDF [561](#)
 HLQDDDF command
 DIRECTION [562](#)
 HLQDDDF command syntax [561](#)
 HLQDDDF, table space cloning [561](#)

I

IDCAMS-DDN, table space cloning [528](#)
 IGNORE-CREATE-OBJECT-EXISTS, table space cloning [545](#)
 IGNORE-DROP-OBJECT-DOES-NOT-EXIST, table space cloning [545](#)
 IGNORE-RF-MISMATCH-IF-NO-VAR-COLS, table space cloning [529](#)
 image copies, using as cloning source [289](#), [290](#)
 IMAGE-COPY-PREFERENCE, table space cloning [532](#)
 IMPORT command
 for importing ISPF profiles and control files [101](#)
 INCLUDE-ALL-RI, table space cloning [529](#)
 INDEX-LOG-APPLY, table space cloning [533](#)
 indexes
 dynamic rebuild [297](#)
 intelligent rebuild [297](#)
 rebuilding [297](#)
 recommendations for rebuilding [298](#)
 INI keywords
 SI027_VALUES [1146](#)
 SI040_VALUES [1146](#)
 INIMERGE_VALUES keyword [1146](#)
 installation verification process (IVP) [98](#)
 intelligent rebuild of indexes [297](#)
 INTELLIGENT-REBUILD, table space cloning [570](#)
 IP-VERSION6, table space cloning [571](#)
 ISPF interface
 table space cloning DD descriptions [330](#)
 ISPF interface (classic)
 about [323](#)
 cloning job reference [341](#)
 configuring Db2 subsystems [324](#)
 configuring user settings [326](#)
 creating a cloning profile [335](#)
 creating cloning jobs [334](#)
 setting job card defaults [326](#)
 setting subsystem cloning defaults [326](#)
 setting table space cloning defaults [327](#)
 specifying subsystem cloning information [325](#)
 specifying table space cloning information [325](#)
 subsystem cloning [341](#)
 subsystem cloning steps [335](#)
 table space cloning across multiple profiles [355](#)
 table space cloning default DDs [328](#)
 table space cloning steps [349](#)
 ISPF interface (new)
 application cloning from Db2 source data sets [315](#)
 application cloning from image copies [316](#)
 application cloning profiles, managing [315](#)
 application cloning profiles, working with [313](#)
 application cloning using user copies [316](#)
 application cloning without shared DASD [317](#)
 application profiles [103](#)
 building application cloning jobs [319](#)
 configuring Db2 subsystems [312](#)
 creating application cloning profiles [313](#), [314](#)
 defining job cards for [312](#)
 getting started [311](#)
 migrating table space cloning profiles to application cloning profiles [103](#)
 selecting source objects [318](#), [319](#)
 selecting the source and target Db2 SSIDs [318](#)

ISPF interface (new) (*continued*)
 submitting jobs for application cloning from Db2 source data sets [321](#)
 submitting jobs for application cloning from image copies [321](#)
 submitting jobs for application cloning from user copies [321](#)
 submitting jobs for application cloning without shared DASD [322](#)
 verifying DD specifications for the cloning jobs [320](#)

ISPF profiles
 exporting [100](#)
 importing [101](#)
 migrating to application cloning profiles [103](#)

ISSUE_CKZ14141I keyword [1146](#)
 ISSUE_VCLOSE keyword [1146](#)
 ISSUE_VCLOSE_SCOPE keyword [1146](#)
 ISSUE-UCAT-UNALLOCATE
 RENAME command [481](#)

ISSUE-VCLOSE
 RENAME command [481](#)

IVP [98](#)

J

JCL
 BACKINFO-REFORMAT [376](#)
 BCSCLEAN [382](#)
 COPY [402](#)
 COPY-BY-DS [411](#)
 COPYCHECK [414](#)
 DB2ALTERBSDS [418](#)
 DB2ALTERBSDS step
 Sample control statements [419](#)
 DB2FIX [424](#)
 DB2GETBACKINFO [428](#)
 DB2LGRNXCLEAN [431](#)
 DB2RBLDBSDS [434](#)
 DB2SCHEMA-UPDATE [439](#)
 DB2SETLOG [441](#)
 DB2SETLOG step
 Sample control statements [442](#)
 DB2SQL [447](#)
 DB2START [451](#)
 DB2STOP [454](#)
 DB2UPDATE [460](#)
 DB2UPDATE step
 Sample control statements [461, 462](#)
 DB2UTILXCLEAN [465](#)
 DB2UTILXCLEAN identify objects with UT status [465](#)
 DB2XCFCLEAN [468](#)
 FINDUCATS [469](#)
 JRNLUPGRADE [472](#)
 ONLINECLIP [474](#)
 RENAME [488](#)
 RESTORE-FROM-DUMPTAPES [495](#)
 UCATOPTIONS [498](#)
 VARYOFF [501](#)
 VARYON [506](#)
 VOLOPTIONS [510](#)

job template variables [303, 305](#)
 job templates [303](#)
 job templates for cloning consistent FlashCopy data sets to Db2 data sets using Sterling Connect:Direct [307, 309](#)

job templates for utilities [306](#)
 JOB-TEMPLATE, table space cloning [529](#)
 Journal data set
 Upgrade [470](#)

JOURNAL-DDN
 COPY-BY-DS [408](#)

JOURNAL-DDSN
 COPY [388](#)
 COPYCHECK [413](#)
 DB2ALTERBSDS [416](#)
 JRNLUPGRADE [471](#)
 ONLINECLIP [473](#)
 RENAME [477](#)
 VARYOFF [500](#)
 VARYON [504](#)
 VOLOPTIONS [509](#)

JOURNAL-DSN
 COPY [388](#)
 COPY-BY-DS [408](#)
 COPYCHECK [413](#)
 DB2ALTERBSDS [416](#)
 DB2LGRNXCLEAN [431](#)
 DB2RBLDBSDS [433](#)
 JRNLUPGRADE [471](#)
 ONLINECLIP [473](#)
 RENAME [477](#)
 VARYOFF [500](#)
 VARYON [504](#)
 VOLOPTIONS [509](#)

JRNLUPGRADE
 Command scenarios [470](#)

JRNLUPGRADE command
 Command and keyword definitions [471](#)
 JCL [472](#)
 JOURNAL-DDN [471](#)
 JOURNAL-DSN [471](#)
 Keyword definitions [471](#)
 OLD-JOURNAL-DDN [471](#)
 OLD-JOURNAL-DSN [471](#)
 JRNLUPGRADE command syntax [471](#)

K

KEEP-DATABASES-ON-DISCONNECT, table space cloning [571](#)
 KEEP-VOLUMES-SEQUENCE
 RESTORE-FROM-DUMPTAPES command [492](#)

Keyword definitions
 BCSCLEAN command [380](#)
 COPY
 JOURNAL-DSN [388](#)
 COPY command [388](#)
 COPY-BY-DS
 JOURNAL-DSN [408](#)
 COPY-BY-DS command [407](#)
 COPYCHECK
 JOURNAL-DSN [413](#)
 DB2ALTERBSDS
 JOURNAL-DSN [416](#)
 FINDUCATS command [469](#)
 JRNLUPGRADE
 JOURNAL-DSN [471](#)
 OLD-JOURNAL-DSN [471](#)
 JRNLUPGRADE command [471](#)

Keyword definitions (*continued*)

ONLINECLIP
 JOURNAL-DSN [473](#)
ONLINECLIP command [473](#)
RENAME
 JOURNAL-DSN [477](#)
RENAME command [477](#)
VARYOFF
 JOURNAL-DSN [500](#)
VARYOFF command [500](#)
VARYON
 JOURNAL-DSN [504](#)
VARYON command [504](#)
VOLOPTIONS
 JOURNAL-DSN [509](#)
VOLOPTIONS command [508](#)

L

LA-ENABLE, table space cloning [531](#)
LAST [427](#)
Launchpad, adding Db2 Cloning Tool [99](#)
legal notices
 cookie policy [1183](#), [1185](#)
 notices [1183](#)
 programming interface information [1183](#)
 trademarks [1183](#), [1184](#)
LIST
 DB2GETBACKINFO [427](#)
 UCATOPTIONS command [497](#)
 VOLOPTIONS command [509](#)
LISTDEF [562](#)
LISTDEF command syntax [564](#)
LISTDEF INCLUDE/EXCLUDE, table space cloning [565](#)
LISTDEF, table space cloning [565](#)
load libraries
 APF authorizing [30](#)
LOCAL-SSID, table space cloning [568](#)
LOCATION
 DB2GETBACKINFO [426](#)
LOG-APPLY [293](#)
LOG-APPLY options for rebuilding indexes [297](#)
LOG-APPLY source job changes [294](#)
LOG-APPLY source TCP/IP server job changes [294](#)
LOG-APPLY target job changes [296](#)
LOG-APPLY, table space cloning [530](#)
logs, applying to target objects [293](#)
LOGSONLY
 DB2GETBACKINFO [427](#)
LONGVAR-COMPATIBILITY, table space cloning [540](#)

M

maintenance [107](#)
masks
 specifying for DB2UPDATE STOGROUPS keyword [456](#)
MAX_COPY_RC keyword [1146](#)
MAX_RC keyword [1146](#)
MAX_RENAME_TASKS keyword [1146](#)
MAX_SUBTASKS keyword [1146](#)
MAX-CONCURRENT-CMDS
 DB2FIX [422](#)
MAX-COPY-RC, table space cloning [571](#)

MAX-RC, table space cloning [572](#)
MAX-SUBTASKS, table space cloning [572](#)
MAX-TAPEDRIVES
 RESTORE-FROM-DUMPTAPES command [492](#)
MAX-TASKS
 RENAME command [482](#)
MAX-VOLS-PER-CMD
 VARYOFF command [500](#)
MAX-VOLS-PER-CMD command
 VARY-DDN [505](#)
MEMBERS-AND-DBD01
 DB2FIX [423](#)
MEMBERS-NEED-STARTING
 DB2FIX [423](#)
MERGE-PRINT, table space cloning [572](#)
message conventions [583](#)
message explanations, methods for accessing [1080](#)
metadata library
 overview [1087](#)
 specifying [72](#)
MGMTCLAS
 RENAME command [482](#)
MGMTCLAS-PAIRS
 RENAME command [483](#)
MGMTCLAS, table space cloning [526](#)
migrated application data set [107](#)
MIGRATED-DSN
 COPY-BY-DS command [410](#)
MIM_GDIF keyword [1146](#)
MINILOG-LARGE-FILE-TYPE, table space cloning [533](#)
mirror image [14](#)
MISSING_USERCAT_DISP keyword [1146](#)
MISSING_USERCAT_RC keyword [1146](#)
MISSINGUCAT
 RENAME command [483](#)
modifying settings [72](#)
MODLEVEL [1082](#)
multivolume data sets [107](#)

N

NEWCATWORKS
 UCATOPTIONS command [498](#)
NEWCATWORKS-DDN
 UCATOPTIONS command [498](#)
Newname masks
 Syntax [476](#)
NEWTARGETS
 UCATOPTIONS command [498](#)
 VOLOPTIONS command [510](#)
NEWTARGETS-DDN
 UCATOPTIONS command [498](#)
 VOLOPTIONS command [510](#)
NEWTARGETSDEVN
 VOLOPTIONS command [510](#)
non-VSAM data set aliases
 rename masks [107](#)
NOT_RENAMED_DISP keyword [1146](#)
NOT_RENAMED_RC keyword [1146](#)
notices [1183](#)
NOTRENAMED
 RENAME command [483](#)
NOUSERCATALOGS
 COPY command [391](#)

NULLSTORCLAS, table space cloning [526](#)

O

OBJECT-MISMATCH-RETURN-CODE, table space cloning [540](#)

OBJECT-TRANSLATE

considerations for [554](#)

OBJECT-TRANSLATE, table space cloning [542](#)

OFFLINE

COPY-BY-DS command [410](#)

offline cloning [126](#)

offline cloning steps [127](#), [129](#), [132](#)

offline cloning steps summary [126](#), [128](#), [131](#)

Offline volume

With DATA-MOVER(PGM(ADRDSSU)) [401](#)

OLD-JOURNAL-DDN

JRNLPGRADE [471](#)

OLD-JOURNAL-DSN

JRNLPGRADE [471](#)

Oldname masks

Syntax [476](#)

Oldvalue masks

Syntax [374](#), [436](#)

online cloning [134](#)

online cloning steps [137](#)

online cloning steps summary [135](#)

online cloning steps summary, remove data sharing members [140](#)

online cloning steps summary, target becoming non-data sharing [146](#)

online cloning steps, removal of data sharing members [142](#)

online cloning steps, target becoming non-data sharing [148](#)

online status of target volumes [107](#)

ONLINECLIP [472](#)

ONLINECLIP command

Command and keyword definitions [473](#)

JCL [474](#)

JOURNAL-DDN [473](#)

JOURNAL-DSN [473](#)

Keyword definitions [473](#)

WRONG-VOLSER [473](#)

ONLINECLIP command syntax [473](#)

onsite mirror [119](#)

ORPHAN_CATENTRY_DISP keyword [1146](#)

ORPHAN_CATENTRY_RC keyword [1146](#)

ORPHANCATENTRY

RENAME command [484](#)

overview [1](#)

P

pairing source and target volumes [107](#)

panels

Copy DB2 Entries [92](#)

parameters

customization [31](#)

PARTLEVEL, table space cloning [566](#)

pattern masks, filtering [26](#)

pervasive data encryption [111](#)

planning vol copies and renames

discrete vols vs. SMS storage group [107](#)

populated ICF catalog [110](#)

POST request

for table space cloning via REST API [366](#)

preparing to use Tools Customizer [72](#)

PRESMIRNONE, table space cloning [525](#)

PRESMIRPREF, table space cloning [525](#)

PRESMIRREQ, table space cloning [525](#)

problems

diagnostic information about [1080](#), [1081](#), [1083](#)

procedures

cloning Db2 subsystem [112](#)

PROCESS-DDL-DDN, table space cloning [544](#)

PROCESS-DDL, table space cloning [543](#)

PROCESS-DDLDDL - ATTRIBUTE - CHANGE

DDL-ATTRIBUTE-CHANGE [555](#)

PROCESS-TYPE, table space cloning [543](#)

PROCESS-UNMATCHED-TARGET-INDEXES, table space cloning [572](#)

processing DDL [195](#)

product authorizations [97](#)

PRODUCT_INFO keywords [1146](#)

profiles

exporting and importing using ISPF interface (new and classic) [99](#)

migrating to application cloning profiles [103](#)

programming interface information [1183](#)

PUT request

for table space cloning via REST API [366](#)

Q

QUIESCE-POINT, table space cloning [535](#)

R

RACF authorities [109](#)

READ-FROM-SERVER-CACHE, table space cloning [521](#)

reader comment form [16](#)

REBUILD-INDEXES-EXECUTE, table space cloning [573](#)

REBUILD-INDEXES-REPORT, table space cloning [573](#)

REBUILD-UNMATCHED-TARGET-INDEXES, table space cloning [574](#)

rebuilding indexes [297](#), [298](#)

RECATALOG

RENAME command [484](#)

target catalog renaming considerations [110](#)

RECATALOG keyword [1146](#)

RECREATE-OBJECTS, table space cloning [545](#)

RECREATE-TABLE-RESTRICTED-ON-DROP, table space cloning [545](#)

REMOTE-CONNECT-TYPE, table space cloning [574](#)

remove data sharing members [128](#)

removed functions and parameters [10](#)

RENAME [474](#)

RENAME command

Command and keyword definitions [477](#)

DATACLAS [478](#)

DATACLAS-PAIRS [478](#)

DEFAULT_IF_NO_MATCH [478](#), [487](#)

DRIVEACS [478](#)

EXCLUDE-SRCNAME [479](#)

EXCLUDE-SRCNAME-MASKS [479](#)

EXCLUDE-SRCNAME-MASKS-DDN [479](#)

GDG-ALL-MIGRATED [480](#)

RENAME command *(continued)*

- GDG-EMPTY [480](#)
- GDG-MIGRATED [480](#)
- GDG-TAPE [480](#)
- ISSUE-UCAT-UNALLOCATE [481](#)
- ISSUE-VCLOSE [481](#)
- JCL [488](#)
- JOURNAL-DDN [477](#)
- JOURNAL-DSN [477](#)
- Keyword definitions [477](#)
- MAX-TASKS [482](#)
- MGMTCLAS [482](#)
- MGMTCLAS-PAIRS [483](#)
- MISSINGUCAT [483](#)
- NOTRENAMED [483](#)
- ORPHANCATENTRY [484](#)
- RECATALOG [484](#)
- RENAME-AUDIT-LOG [484](#)
- RENAME-ERROR [485](#)
- RENAME-LIST [485](#)
- RENAME-MASKS [477](#)
- RENAME-MASKS-DDN [478](#)
- RERUN [485](#)
- SAFE [486](#)
- SIMULATE [486](#)
- SOURCE [487](#)
- SPEED [486](#)
- STORCLAS [487](#)
- STORCLAS-PAIRS [487](#)
- TEMPDSN [487](#)
- VOLBKUP-DDN [488](#)
- RENAME command syntax [474](#)
- Rename considerations
 - Filter masks [475](#)
 - Rename masks [475](#)
- rename masks
 - GDG base names [107](#)
 - non-VSAM data set aliases [107](#)
 - VSAM cluster names [107](#)
- Rename masks
 - for COPY-BY-DS [405](#)
 - Newname syntax [476](#)
 - Oldname syntax [476](#)
 - Oldvalue syntax [374](#)
 - Rename considerations [475](#)
- rename masks and existing naming conventions [109](#)
- RENAME_ERROR keyword [1146](#)
- RENAME_ERROR_CONTINUE_RC keyword [1146](#)
- RENAME-AUDIT-LOG
 - RENAME command [484](#)
- RENAME-ERROR
 - RENAME command [485](#)
- RENAME-LIST
 - RENAME command [485](#)
- RENAME-MASKS
 - COPY-BY-DS command [408](#)
 - RENAME command [477](#)
- RENAME-MASKS-DDN
 - COPY-BY-DS command [408](#)
 - RENAME command [478](#)
- REPLACE-UNCONDITIONAL
 - COPY-BY-DS command [411](#)
- REPORT-JOB, table space cloning [574](#)
- requirements

requirements *(continued)*

- source and target volume pairing [107](#)
- RERUN
 - RENAME command [485](#)
 - RESTORE-FROM-DUMPTAPES command [492](#)
- RESET_LOGRBA keyword [1146](#)
- RESET-LOGRBA, table space cloning [546](#)
- REST API, table space cloning with
 - configuring and running the workflow [365](#)
 - JSON multi-line variables values for [364](#)
 - multi-line variables values for [363](#)
 - running the workflow via REST services [366](#)
 - running the workflow via the z/OSMF web browser-based interface [368](#)
 - sample members [357](#)
 - variable input file parameters [358](#)
- RESTORE-FROM-DUMPTAPES [489](#)
- RESTORE-FROM-DUMPTAPES command
 - BACKINFO-DDN [491](#)
 - Command and keyword definitions [491](#)
 - EXCLUDE-TO-VOLSER [492](#)
 - JCL [495](#)
 - KEEP-VOLUMES-SEQUENCE [492](#)
 - MAX-TAPEDRIVES [492](#)
 - RERUN [492](#)
 - SIMULATE [493](#)
 - SOURCE-STORAGEGROUP [493](#)
 - STATUS-DDN [493](#)
 - TAPE-UNIT [493](#)
 - TARGET-UCATS-ON-TARGET-VOLUMES [493](#)
 - TO-STORAGEGROUP [491](#)
 - TO-VOLSER [491](#)
 - USERCATALOGS [494](#)
 - USERCATALOGS-DDN [494](#)
 - VARY-SCOPE [494](#)
 - VOLPAIRS-DDN [494](#)
- RESTORE-FROM-DUMPTAPES command syntax [491](#)
- RESTRICT-STATUS-VALUES, table space cloning [574](#)
- RESUME
 - VOLOPTIONS command [509](#)
- return code
 - changing value [111](#)
- REUSE, table space cloning [526](#)
- RI parameter, table space cloning [566](#)
- roadmaps
 - customizing for the first time [77](#)
 - first-time customization [77](#)
- RTS-COPY-ENABLE, table space cloning [546](#)
- RTS-COPY, table space cloning [546](#)
- RTS-DETAILS, table space cloning [547](#)
- RTSFILE-DATA-SET-HLQ, table space cloning [547](#)
- RVA DASD
 - RVA partition [107](#)
 - source and target volume pairing [107](#)
 - track format [107](#)

S

- SAFE
 - RENAME command [486](#)
- SCAN-ONLY, table space cloning [575](#)
- schema masks
 - Oldvalue syntax [436](#)
- SCOPE

SCOPE (*continued*)
 VARYOFF command [500](#)
 SCOPE command
 VARY-DDN [505](#)
 screen readers and magnifiers [16](#)
 security [97](#)
 selection of source/target volumes
 discrete vols vs. SMS storage group [107](#)
 service information [15](#)
 SET [567](#)
 SET command syntax [567](#)
 SET, table space cloning [568](#)
 SI027_VALUES [1146](#)
 SI040_VALUES [1146](#)
 SIMULATE
 BCSCLEAN command [381](#)
 COPY command [400](#)
 COPY-BY-DS command [411](#)
 DB2FIX [423](#)
 RENAME command [486](#)
 RESTORE-FROM-DUMPTAPES command [493](#)
 VARYOFF command [501](#)
 VOLOPTIONS command [509](#)
 SIMULATE command
 VARY-DDN [505](#)
 SIMULATE, table space cloning [548](#)
 SKIP-LOG-APPLY, table space cloning [535](#)
 SKIP-SYSLGRNX, table space cloning [535](#)
 SMS class values [443](#)
 SnapShot
 source and target volume pairing [107](#)
 SOURCE
 RENAME command [487](#)
 VARYOFF command [500](#)
 VARYON command [504](#)
 source and target volume
 migrated application data set [107](#)
 source and target volume pairing [107](#)
 source and target volumes
 pairing and requirements [107](#)
 source volume
 condition [107](#)
 data set/sphere integrity [107](#)
 Source volume
 Names [383](#)
 Source volumes
 Offline volumes [401](#)
 SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT, table space
 cloning [576](#)
 SOURCE-IP-VERSION6, table space cloning [576](#)
 SOURCE-STORAGEGROUP
 RESTORE-FROM-DUMPTAPES command [493](#)
 SOURCE-TCP-SERVER-JOB, table space cloning [576](#)
 SOURCE-TCPIP-SERVER-IP, table space cloning [576](#)
 SOURCE-TCPIP-SERVER-NAME, table space cloning [576](#)
 SOURCE-TCPIP-SERVER-PORT, table space cloning [577](#)
 SOURCE-TCPIP-STC-NAME, table space cloning [577](#)
 SOURCECLIP
 VOLOPTIONS command [509](#)
 SOURCESONLINE
 COPY command [400](#)
 SPACE_MANAGEMENT [1146](#)
 SPACES-PER-LOG-APPLY-CALL, table space cloning [537](#)
 SPACES-PER-MINILOG, table space cloning [537](#)
 SPECIAL_SECTIONS keyword [1146](#)
 specifying data sets [72](#)
 SPEED
 RENAME command [486](#)
 START-SCOPE
 DB2FIX [423](#)
 START-SOURCE-DDN, table space cloning [549](#)
 status of inflight transactions [135](#)
 STATUS-DDN
 RESTORE-FROM-DUMPTAPES command [493](#)
 STOGROUPS keyword considerations for subsystem cloning
[456](#)
 STOP-SOURCE-DDN, table space cloning [549](#)
 STOP-TARGET-DDN, table space cloning [549](#)
 STOPSNAP
 COPYCHECK [414](#)
 storage group specification [107](#)
 Storage group specification
 DATA-MOVER(PGM(ADRDSSU)) [402](#)
 storage management [107](#)
 STORCLAS
 RENAME command [487](#)
 STORCLAS-PAIRS
 RENAME command [487](#)
 STORCLAS, table space cloning [526](#)
 stored procedure
 about [153](#)
 building the verified cloning [170](#)
 calling with a REXX EXEC [168](#)
 example of cloning a non-data sharing DB2 system [172](#)
 example of cloning a non-data sharing DB2 system by
 data set [173](#)
 example of cloning a non-data sharing DB2 system from
 a DB2 system-level backup [174](#)
 example of using the stored procedure to clone a non-
 data sharing subsystem from DB2 system-level backup
 dump tapes [175](#)
 example of using the stored procedure to clone a non-
 data sharing subsystem from DB2 system-level backup
 dump tapes across sysplexes [176](#)
 JCL members [169](#)
 migrating [154](#)
 parameter files and parameter descriptions [155](#)
 recloning a DB2 system [171](#)
 results set [171](#)
 REXX EXEC, calling stored procedure from [168](#)
 running the verified cloning [170](#)
 steps for [155](#)
 syntax [167](#)
 verifying the cloning [169](#)
 stored procedure, using [153](#)
 subsystem cloning
 add/verify high level qualifiers [338](#)
 building the cloning jobs [340](#)
 ISPF interface (classic) [323–328](#), [334–341](#), [349–353](#),
[355](#)
 planning for [107](#)
 selecting source and target subsystems [337](#)
 selecting source and target subsystems (data sharing)
[337](#)
 selecting source and target subsystems (non-data
 sharing) [337](#)
 specify rename masks [339](#)
 specify source and target ICF catalogs [339](#)

- subsystem cloning (*continued*)
 - specify source and target volume pairing [339](#)
 - submitting the cloning jobs [349](#)
- subsystem cloning profiles
 - exporting and importing [99–101](#)
- subsystem cloning step summary
 - ISPF interface (classic) [323–328](#), [334–341](#), [349–353](#), [355](#)
- subsystem cloning to a specific point in time from a BACKUP SYSTEM backup [1115](#)
- subsystem cloning using SLB dump tapes [1128](#)
- subsystem, cloning [112](#)
- SUBTASK-DATASET-EXTENSIONS, table space cloning [577](#)
- SUBTASK-TERMINATION-WAIT, table space cloning [577](#)
- summary of changes [1](#), [1161](#)
- support
 - required information [1080](#), [1081](#), [1083](#)
- support information [15](#)
- supported volume copy products
 - Flash Copy [107](#)
 - FlashCopy [14](#)
 - SnapShot [14](#), [107](#)
- SUPPRESS-GRANTS, table space cloning [545](#)
- SUPPRESS-RI-CONSTRAINTS, table space cloning [545](#)
- SYNCDB2-DDN, table space cloning [549](#)
- syntax
 - keywords in CKZINI member [1146](#)
- Syntax
 - BACKINFO-REFORMAT command [373](#)
 - BCSCLEAN command [379](#)
 - COPY command [386](#)
 - COPY command, table space cloning [514](#)
 - COPY-BY-DS command [407](#)
 - COPYBYDS command
 - RENAME considerations for [405](#)
 - COPYCHECK command [412](#)
 - DATASUBTYPE command [560](#)
 - DB2ALTERBSDS command [415](#)
 - DB2FIX command [421](#)
 - DB2GETBACKINFO command [425](#)
 - DB2LGRNXCLEAN command [430](#)
 - DB2RBLDBSDS command [432](#)
 - DB2SCHEMA-UPDATE command [436](#)
 - DB2SETLOG command [440](#)
 - DB2START command [449](#)
 - DB2STOP command [452](#)
 - DB2UPDATE command [455](#)
 - DB2UTILXCLEAN command [464](#)
 - DB2XCFCLEAN command [466](#)
 - DBSQL command [443](#)
 - FINDUCATS command [468](#)
 - HLQDDDF command [561](#)
 - JRNLUPLGRADE command [471](#)
 - LISTDEF command [564](#)
 - Newname masks [476](#)
 - Oldname masks [476](#)
 - Oldvalue masks [374](#), [436](#)
 - ONLINECLIP command [473](#)
 - RENAME command
 - considerations for [475](#)
 - RESTORE-FROM-DUMPTAPES command [491](#)
 - SET command [567](#)
 - UCATOPTIONS command [497](#)
 - VARYOFF command [499](#)

- Syntax (*continued*)
 - VARYON command [503](#)
 - VOLOPTIONS command [508](#)
- syntax rules, CKZINI member [1145](#)

T

- table space cloning
 - APF authorization [186](#)
 - applying logs during [293](#)
 - building the table space cloning jobs [353](#)
 - caching indexes using catalog prefetch [202](#)
 - considerations for copying versioned objects [204](#)
 - considerations for FUZZY-copy [201](#)
 - considerations for in-progress read/write activity or Db2 utilities [183](#)
 - considerations for renaming data sets [201](#)
 - considerations for target objects created using DEFINE NO [186](#)
 - copy procedure for all other copy methodologies [265](#), [267](#)
 - copy procedure for FlashCopy, SnapShot, or TimeFinder/Clone [261](#), [262](#)
 - copy products used [179](#)
 - DFSMS data encryption [187](#)
 - DFSMS pervasive encryption [187](#)
 - dropped and altered tables [204](#)
 - editing the source job [351](#)
 - encryption, pervasive [187](#)
 - fast replication tools [179](#)
 - function overview [207](#)
 - high level qualifiers [201](#)
 - how table spaces and index spaces are stopped and started [190](#), [191](#)
 - identity columns [184](#)
 - implicitly created object [184](#)
 - ISPF interface (classic) [323–328](#), [334–341](#), [349–353](#), [355](#)
 - LOBs, BLOBs, CLOBs, and DBCLOBs [185](#)
 - missing target Db2 objects [186](#)
 - non-SMS managed volumes [189](#)
 - object attributes [192](#)
 - object compatibility checking [192](#)
 - object mismatch processing [191](#)
 - overview [179](#)
 - pervasive data encryption [187](#)
 - planning for catalog prefetch [201](#)
 - planning for cloning [183](#)
 - plans and packages [204](#)
 - RACF and ACF authorities [186](#)
 - renaming table spaces and index spaces [183](#)
 - return code choices [204](#)
 - rules for using [183](#)
 - selecting source and target subsystems [350](#)
 - source job overview [207](#)
 - space processing order [190](#)
 - strategies for resolving object mismatches [194](#)
 - submitting the cloning jobs [355](#)
 - subsystem copy rules [183](#)
 - target data set ICF catalog aliases [201](#)
 - target job overview [207](#)
 - TCP/IP considerations [186](#)
 - TCP/IP job overview [208](#)
 - TCP/IP server job overview [208](#)

table space cloning (*continued*)

- tobject mismatch processing [194](#)
- verify the report job settings [352](#)
- verify the source TCP/IP server job settings [353](#)
- verify the TCP/IP server job settings [353](#)
- verifying the target job settings [352](#)
- XML [185](#)

table space cloning profiles

- exporting and importing [99–101](#)

table space cloning step summary

- ISPF interface (classic) [323–328](#), [334–341](#), [349–353](#), [355](#)

table space cloning via REST API

- configuring and running the workflow [365](#)
- JSON multi-line variables values for [364](#)
- multi-line variables values for [363](#)
- running the workflow via REST services [366](#)
- running the workflow via the z/OSMF web browser-based interface [368](#)
- sample members for [357](#)
- variable input file parameters [358](#)

TAPE-UNIT

- RESTORE-FROM-DUMPTAPES command [493](#)

TARGET

- VARYOFF command [500](#)
- VARYON command [504](#)

target becoming non-data sharing [131](#), [132](#)

Target catalog

- Delete previous application cycle entries using BCSCLEAN [379](#)

target catalog renaming considerations

- RECATALOG option [110](#)

target objects, re-creating [196](#)

target volume

- condition [107](#)
- data set/sphere integrity [107](#)

Target volume names [383](#)

target volumes

- online status [107](#)

Target volumes

- Offline volumes [402](#)

TARGET_VOLS_SHOULD_BE_EMPTY keywords [1146](#)

TARGET-DB2, table space cloning [517](#)

TARGET-JOB-INDEX-REBUILD-DDN, table space cloning [549](#)

TARGET-JOB-MAX-RC, table space cloning [578](#)

TARGET-JOB-REPAIR-DDN, table space cloning [550](#)

TARGET-JOB-REPAIR-EXECUTE, table space cloning [578](#)

TARGET-JOB, table space cloning [578](#)

TARGET-PREFETCH-DATABASE-LIST, table space cloning [521](#)

TARGET-UCATS-ON-TARGET-VOLUMES

- COPY command [401](#)
- RESTORE-FROM-DUMPTAPES command [493](#)

TARGET-VOLS-SHOULD-BE-EMPTY

- COPY command [401](#)

TARGETOFFLINECLIP

- VOLOPTIONS command [509](#)

TARGETSONLINE

- COPY command [401](#)

TARGETSUONLINE

- COPY command [401](#)

TCP-SERVER-JOB, table space cloning [579](#)

TCP/IP server job [208](#)

TCPIP_SERVER_PORT keyword [1146](#)

TCPIP_STC_NAME keyword [1146](#)

TCPIP-SERVER-IP, table space cloning [579](#)

TCPIP-SERVER-NAME, table space cloning [579](#)

TCPIP-SERVER-PORT, table space cloning [579](#)

TCPIP-STC-NAME, table space cloning [579](#)

technotes [16](#)

TEMP_DATASET_DISP keyword [1146](#)

TEMP_DATASET_RC keyword [1146](#)

TEMPDSN

- RENAME command [487](#)

TEMPLATE DDs for UNLOAD-LOAD, table space cloning [333](#), [334](#), [552](#)

TEMPLATE-SORTOUT-DDN, table space cloning [552](#)

TEMPLATE-VARIABLE, table space cloning [580](#)

terminology for "objects" [181](#)

TO-STORAGEGROUP

- COPY command [392](#)
- RESTORE-FROM-DUMPTAPES command [491](#)

TO-USER-STORAGEGROUP

- COPY command [393](#)

TO-VOLSER

- COPY command [393](#), [394](#)
- RESTORE-FROM-DUMPTAPES command [491](#)

TOKEN

- DB2GETBACKINFO [427](#)

Tools Customizer

- associated list
 - adding Db2 entries [83](#)
 - overview [1085](#)
- associating Db2 entries [83](#)
- browsing parameters [92](#)
- component [1085](#)
- Copy DB2 Entries panel [92](#)
- copying Db2 entries [92](#)
- Create a DB2 Entry panel [83](#)
- creating Db2 entries [83](#)
- customization jobs
 - deleting [95](#)
 - displaying [95](#)
 - generating [90](#)
 - maintaining [95](#)
 - renaming [95](#)
 - sort sequence [90](#)
 - submitting [90](#), [95](#)
- customization library
 - deleting jobs [95](#)
 - maintaining [95](#)
 - recustomizing [95](#)
 - renaming jobs [95](#)
- customization library qualifier
 - specifying [72](#)
- Customized status [1085](#)
- Customizer Workplace panel [90](#)
- customizing a new version of a product [77](#), [78](#)
- customizing a product for the first time [77](#)
- data sets
 - customization library [1087](#)
 - data store [1087](#)
 - Discover EXEC library [1087](#)
 - metadata library [1087](#)
- data store data set
 - specifying [72](#)
- Db2 data sharing members
 - adding [83](#)

Tools Customizer (*continued*)

- Db2 data sharing members (*continued*)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
- Db2 entries
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
 - defining [90](#)
 - deleting [94](#)
 - generating jobs for [90](#)
 - removing [94](#)
 - selecting [90](#)
 - specifying [90](#)
 - unassociating [94](#)
- DB2 entries [1085](#)
- Db2 group attach field
 - specifying [72](#)
- Db2 group attach names
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
- Db2 parameters
 - defining [88](#)
 - editing [88](#)
- DB2 Parameters panel [88](#)
- Db2 subsystems
 - adding [83](#)
 - associating [83](#)
 - copying [92](#)
 - creating [83](#)
- DBb2 entries
 - deleting [94](#)
 - unassociating [94](#)
- defining Db2 parameters [88](#)
- defining LPAR parameters [87](#)
- defining parameters [85](#), [90](#)
- defining product parameters [85](#)
- deleting Db2 entries [94](#)
- deleting jobs [79](#)
- Discover Customized Product Information panel [81](#)
- Discover EXEC
 - customizing a new version of a product [77](#), [78](#)
 - retrieving product information automatically [81](#)
- Discovered status [1085](#)
- discovering product information [81](#)
- displaying jobs [95](#)
- editing LPAR parameters [87](#)
- editing product parameters [85](#)
- Errors in Customization status [1085](#)
- finding trace data set [1084](#)
- Finish Product Customization panel [90](#)
- first-time customization [77](#)
- generating jobs [90](#)
- high-level qualifier [1085](#)
- Incomplete status [1085](#)
- job sort order [90](#)
- jobs
 - deleting [95](#)
 - displaying [95](#)
 - maintaining [95](#)

Tools Customizer (*continued*)

- jobs (*continued*)
 - renaming [95](#)
 - submitting [95](#)
- LPAR Parameters panel [87](#)
- maintaining jobs [95](#)
- master list
 - adding Db2 entries [83](#)
 - Associate DB2 Entry for Product panel [83](#)
 - overview [1085](#)
- metadata libraries
 - specifying [80](#)
- metadata library
 - specifying [72](#)
- multiple instances [72](#)
- multiple-LPAR environment [96](#)
- Not Required status [1085](#)
- panels
 - Associate DB2 Entry for Product [83](#)
 - Copy DB2 Entries [92](#)
 - Create a DB2 Entry [83](#)
 - Customizer Workplace [90](#)
 - DB2 Parameters [88](#)
 - Discover Customized Product Information [81](#)
 - Finish Product Customization [90](#)
 - LPAR Parameters [87](#)
 - Product Parameters [85](#)
 - Specify the Metadata Library [80](#)
- parameters
 - browsing [92](#)
 - viewing [92](#)
- preparing to use [72](#)
- product [1085](#)
- product parameters
 - changing [79](#)
 - editing [79](#)
 - modifying [79](#)
- Product Parameters panel [85](#)
- Ready to Customize status [1085](#)
- recustomization [77](#)
- recustomizing a product [77](#), [79](#)
- removing Db2 entries [94](#)
- roadmaps
 - customizing a new version of a product [78](#)
 - recustomizing a product [79](#)
 - using the Discover EXEC [78](#)
- Specify the Metadata Library panel [80](#)
- specifying metadata libraries [80](#)
- starting [71](#)
- status types
 - Customized [1085](#)
 - Discovered [1085](#)
 - Errors in Customization [1085](#)
 - Incomplete [1085](#)
 - Not Required [1085](#)
 - Ready to Customize [1085](#)
- submitting jobs [90](#)
- terminology [1085](#)
- trace data set [1084](#)
- troubleshooting [1083](#)
- user job card settings
 - specifying [72](#)
- using the Discover EXEC [78](#)
- viewing parameters [92](#)

trace data set
 finding [1084](#)
trademarks [1183](#), [1184](#)
transactions in flight [135](#)

U

U.S.A. EBCDIC code set [25](#), [181](#)
UCATOPTIONS [496](#)
UCATOPTIONS command
 BACKUP [497](#)
 BACKUP FORCE [497](#)
 Command and keyword definitions [497](#)
 JCL [498](#)
 LIST [497](#)
 NEWCATWORKS [498](#)
 NEWCATWORKS-DDN [498](#)
 NEWTARGETS [498](#)
 NEWTARGETS-DDN [498](#)
 UPDATE [497](#)
UCATOPTIONS command syntax [497](#)
UNIFIED-WARNING, table space cloning [537](#)
UNLOAD-LOAD-EARLY-VALIDATION , table space cloning
[551](#)
UNLOAD-LOAD-ENABLE, table space cloning [551](#)
UNLOAD-LOAD, table space cloning [550](#)
UPDATE
 UCATOPTIONS command [497](#)
 VOLOPTIONS command [509](#)
UPDATE-DOCID-JCL-DSN, table space cloning [580](#)
USE-DDL-SQLID, table space cloning [546](#)
USE-DUMPTAPES
 DB2GETBACKINFO [428](#)
USE-LAST-CONSISTENT-FLASHCOPY, table space cloning
[526](#)
USE-LOCAL-TIME, table space cloning [537](#)
USE-QUIESCE-POINT-FOR-LOGPOIN, table space cloning
[538](#)
USE-RUNTIME-REPOSITORY, table space cloning [580](#)
USE-TCPIP, table space cloning [538](#)
user job card settings
 specifying [72](#)
user-created clones [119](#)
user-created target volumes [120](#)
USERCATALOGS
 COPY command [388](#)
 DB2GETBACKINFO [428](#)
 RESTORE-FROM-DUMPTAPES command [494](#)
USERCATALOGS-BACKUPFIRST
 COPY command [390](#)
USERCATALOGS-DDN
 RESTORE-FROM-DUMPTAPES command [494](#)
USERCATALOGS-NOBACKUP
 COPY command [389](#)
USERSGDEFS-DDN
 COPY command [394](#)
USERSGDEFS-OFFSETS
 COPY command [394](#)
UTILITY-COMMAND-EXECUTE-PERCENT, table space
cloning [581](#)

V

validating load module contents [1082](#)
VARY-DDN
 MAX-VOLS-PER-CMD command [505](#)
 SCOPE command [505](#)
 SIMULATE command [505](#)
 VARYOFF command [500](#)
 VARYON command [504](#)
 VOL-ALREADY-ONLINE command [505](#)
 WRONG-VOLSER command [506](#)
VARY-SCOPE
 RESTORE-FROM-DUMPTAPES command [494](#)
VARYOFF [499](#)
VARYOFF command
 Command and keyword definitions [500](#)
 COPY-CMD-DDN [500](#)
 JCL [501](#)
 JCL VARYOFF step
 Sample control statements [501](#)
 JOURNAL-DDN [500](#)
 JOURNAL-DSN [500](#)
 Keyword definitions [500](#)
 MAX-VOLS-PER-CMD [500](#)
 SCOPE [500](#)
 SIMULATE [501](#)
 SOURCE [500](#)
 TARGET [500](#)
 VARY-DDN [500](#)
 VOL-ALREADY-OFFLINE [501](#)
VARYOFF command syntax [499](#)
VARYON [503](#)
VARYON command
 Command and keyword definitions [504](#)
 JCL [506](#)
 JOURNAL-DDN [504](#)
 JOURNAL-DSN [504](#)
 Keyword definitions [504](#)
 SOURCE [504](#)
 TARGET [504](#)
 VARY-DDN [504](#)
VARYON command syntax [503](#)
verifying [98](#)
VOL-ALREADY-OFFLINE
 VARYOFF command [501](#)
VOL-ALREADY-ONLINE command
 VARY-DDN [505](#)
VOLBKUP-DDN
 RENAME command [488](#)
VOLOPTIONS [507](#)
VOLOPTIONS command
 Command and keyword definitions [508](#)
 JCL [510](#)
 JOURNAL-DDN [509](#)
 JOURNAL-DSN [509](#)
 Keyword definitions [508](#)
 LIST [509](#)
 NEWTARGETS [510](#)
 NEWTARGETS-DDN [510](#)
 NEWTARGETSDEVN [510](#)
 RESUME [509](#)
 SIMULATE [509](#)
 SOURCECLIP [509](#)
 TARGETOFFLINECLIP [509](#)

- VOLOPTIONS command (*continued*)
 - UPDATE [509](#)
- VOLOPTIONS command syntax [508](#)
- VOLPAIRS
 - COPY command [395](#)
- VOLPAIRS-DDN
 - COPY command [395](#)
 - RESTORE-FROM-DUMPTAPES command [494](#)
- VOLPAIRSDEVN
 - COPY command [395](#)
- VOLPAIRSDEVN-DDN
 - COPY command [395](#)
- VOLPAIRSDEVN-NOCLIP
 - COPY command [395](#)
- VOLPAIRSDEVN-NOCLIP-DDN
 - COPY command [396](#)
- VOLSER mask specification
 - DATA-MOVER(PGM(ADRSSU)) [401](#)
- volume
 - condition [107](#)
- volume cloning
 - planning for [107](#)
 - using multiple interim sets of volumes [1096](#)
- Volume copies
 - COPY step JCL [402](#)
 - COPY-BY-DS step JCL [411](#)
- volume copies and renames
 - data set renaming considerations [109](#)
 - migrated application data sets [107](#)
 - RACF and ACF authorities [109](#)
 - requirements [107](#)
 - target ICF catalog renaming considerations
 - populated or empty catalog [110](#)
 - RECATALOG option of RENAME command [110](#)
- volume copy products supported [107](#)
- Volume names
 - For source and target [383](#)
- volumes
 - SMS storage group [107](#)
- volumes not found [402](#)
- VSAM spheres
 - cluster names matching rename mask [107](#)
 - source volume list [107](#)
- VTOCIX_REBUILDER keyword [1146](#)

- WLM ENVIRONMENT [443](#)
- WORK-DDN
 - DB2GETBACKINFO [426](#)
- WORK-DSN
 - DB2GETBACKINFO [426](#)
- WRONG-VOLSER
 - ONLINECLIP command [473](#)
- WRONG-VOLSER command
 - VARY-DDN [506](#)

Z

- ZPARAM-MEMBER, table space cloning [540](#)

W

- WAIT
 - COPYCHECK [413](#)
 - DB2FIX [423](#)
- WAIT keyword in COPYCHECK step [412](#)
- WAIT-AND-DBD01
 - DB2FIX [424](#)
- WARN-IF-OBJECT-NOT-TRANSLATED, table space cloning [553](#)
- WARN-IF-SKIP-QUIESCE, table space cloning [538](#)
- WARN-IF-TS-DEFINED-LOG-NO, table space cloning [538](#)
- WARN-ON-DATASET-EXTENSION-MISMATCH, table space cloning [553](#)
- WARN-ON-INCOMPLETE-RI, table space cloning [553](#)
- WARN-ON-SIMPLE-TABLESPACE, table space cloning [554](#)
- what's new [1](#), [1161](#)
- WITHDRAW
 - COPYCHECK [413](#)



Product Number: 5655-N15

SC27-6556

