Db2 12 for z/OS

Installation and Migration Guide

IBM
Db2 12 for z/OS

Installation and Migration Guide

IBM
Contents

About this information ................................................................. xi
Who should read this information ................................................. xii
Db2 Utilities Suite for z/OS ............................................................... xii
Terminology and citations ............................................................ xii
Accessibility features for Db2 12 for z/OS ........................................ xiii
How to send your comments ......................................................... xiv
How to read syntax diagrams ....................................................... xiv

Chapter 1. Installing and migrating Db2 ........................................ 1

Chapter 2. Preparing your system to install or migrate to Db2 12 .......... 5
Hardware and software requirements for Db2 .................................... 5
Required maintenance for Db2 12 installation or migration .................. 6
Required authorization for installation or migration ............................ 6
Storage requirements for Db2 ........................................................... 8
  Calculating storage requirements with the installation CLIST ............. 9
  Db2 subsystem storage requirements ............................................. 10
  Virtual storage requirements for address spaces ............................. 19
  Virtual storage requirements for storage pools and working storage .... 25
  Calculating real storage requirements ........................................... 33
Planning for performance of Db2 ..................................................... 35
Planning for release coexistence in Db2 12 ....................................... 36

Chapter 3. Loading the Db2 libraries .............................................. 37
Editing the SMP/E jobs .................................................................. 37
  Creating JOB statements ............................................................ 37
  Choosing link list options ........................................................... 37
  Accessing the correct Db2 program library ...................................... 40
  Performance implications of installation libraries ............................. 41
  Naming conventions for Db2 library names ...................................... 42
  Specifying SMP/E data set options .............................................. 43
  Specifying a new SMP/E prefix for IRLM ...................................... 44
What you produce ......................................................................... 44

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST ............................................. 49
Updating the CLIST defaults input member: job DSNTIJXZ ................. 53
DSNTXAZP tool .............................................................................. 59
The Db2 installation CLIST panel session .......................................... 61
  DSNTIPA1 or DSNTILA1: Main panel ............................................ 64
  DSNTIPO1: OTC license usage panel ............................................. 74
  DSNTIPO2: Notice and acceptance of OTC license panel ................. 74
  DSNTIAP: Provision DB2 - provisioning preferences and data set names 75
  DSNTIAP2: Db2 catalog and directory panel .................................... 77
  DSNTIAP3: Data parameters panel ................................................. 84
  DSNTIAP4: BSDS and active log panel ........................................... 88
  DSNTIPO0: Activate a function level panel ..................................... 91
  DSNTIPK: Define group or member panel ....................................... 96
  DSNTIPG: Installation preferences panel ....................................... 100
  DSNTIPGI: Installation preferences panel 2 .................................... 106
  DSNTIPH: System resource data set names panel ............................ 109
  DSNTIPT: Db2 data set names panel ............................................. 114
  DSNTI PU: Host language data set names panel 1 .......................... 119
  DSNTIPU1: Host language data set names panel 2 ......................... 128
  DSNTIPW: Other data set names panel ....................................... 131
Chapter 5. Installing Db2 ........................................ 413

Installation step 1: Define the SMS environment for Db2 catalog and directory data sets: DSNTIJSS ........................................ 414
Installation step 2: Define Db2 to IRLM and MVS: DSNTIJMV ........................................ 415
Installation step 3: Define Db2 address space startup procedures, language procedures, and the Java WLM environment: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ ........................................ 419
Installation step 4: Define the ICF catalog and alias: DSNTIJCA (optional) ........................................ 421
### Chapter 6. Migrating Db2 to Db2 12

<table>
<thead>
<tr>
<th>Migration step</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration step 1:</td>
<td>Actions to complete before migration</td>
<td>476</td>
</tr>
<tr>
<td>Migration step 2:</td>
<td>Verify the integrity of Db2 table spaces (optional)</td>
<td>478</td>
</tr>
<tr>
<td>Migration step 3:</td>
<td>Identify invalid packages (optional)</td>
<td>485</td>
</tr>
<tr>
<td>Migration step 4:</td>
<td>Check for consistency between catalog tables (optional)</td>
<td>487</td>
</tr>
<tr>
<td>Migration step 5:</td>
<td>Take image copies of the directory and catalog: DSNTIJIC</td>
<td>488</td>
</tr>
<tr>
<td>Migration step 6:</td>
<td>Connect Db2 to TSO</td>
<td>489</td>
</tr>
<tr>
<td>Migration step 7:</td>
<td>Connect Db2 to IMS (optional)</td>
<td>492</td>
</tr>
<tr>
<td>Migration step 8:</td>
<td>Connect CICS to Db2 (optional)</td>
<td>492</td>
</tr>
<tr>
<td>Migration step 9:</td>
<td>Stop Db2 11 activity</td>
<td>492</td>
</tr>
<tr>
<td>Migration step 10:</td>
<td>Back up your Db2 11 volumes (optional)</td>
<td>493</td>
</tr>
<tr>
<td>Migration step 11:</td>
<td>Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM</td>
<td>495</td>
</tr>
<tr>
<td>Migration step 12:</td>
<td>Establish subsystem security (optional)</td>
<td>497</td>
</tr>
<tr>
<td>Migration step 13:</td>
<td>Define Db2 12 to IRLM and MVS: DSNTIJMV</td>
<td>498</td>
</tr>
<tr>
<td>Migration step 14:</td>
<td>update Db2 address space start and language procedure definitions: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ</td>
<td>500</td>
</tr>
<tr>
<td>Migration step 15:</td>
<td>Define system data sets: DSNTIJIN</td>
<td>502</td>
</tr>
<tr>
<td>Migration step 16:</td>
<td>Define user authorization exit routines: DSNTIJEX (optional)</td>
<td>502</td>
</tr>
<tr>
<td>Migration step 17:</td>
<td>Refresh the z/OS definition of Db2</td>
<td>504</td>
</tr>
<tr>
<td>Migration step 18:</td>
<td>Start Db2 12</td>
<td>505</td>
</tr>
<tr>
<td>Migration step 19:</td>
<td>Tailor Db2 12 catalog: DSNTIJTC</td>
<td>507</td>
</tr>
<tr>
<td>Migration step 20:</td>
<td>Ensure that the catalog has no problems (optional)</td>
<td>508</td>
</tr>
<tr>
<td>Migration step 21:</td>
<td>Prepare dynamic SQL program: DSNTIJFM</td>
<td>508</td>
</tr>
</tbody>
</table>
Chapter 7. Automating Db2 installation or migration with z/OS Management Facility

Automating Db2 installation by using z/OS Management Facility ........................................... 563
Automating Db2 migration by using z/OS Management Facility .................................................. 570
Installation panels for installing or migrating with z/OSMF ............................................................... 579
Sample jobs for installing and migrating with z/OSMF ................................................................. 587
  Job DSNTIJDC .......................................................... 587
  Job DSNTIJISO ......................................................... 587
  Job DSNTIJSA .......................................................... 588
  Job DSNTIJMB .......................................................... 588
  DSNTMVSBS tool ....................................................... 588
  Job DSNTIJOS: bring Db2 subsystem parameter changes online ............................................ 593
Sample workflow artifacts for installation and migration with z/OS Management Facility ............... 593

Chapter 8. Creating services for provisioning Db2 for z/OS subsystems (DBaaS) ............................. 599

Chapter 9. Installing, migrating, and enabling Db2 data sharing .................................................. 601
Planning for Db2 data sharing ........................................................................................................... 602
  Parallel Sysplex requirements for Db2 data sharing ......................................................... 602
  Naming in Db2 data sharing ....................................................................................................... 608
  Planning for availability in a data sharing environment ......................................................... 624
  Storage estimates for data sharing environments ..................................................................... 639
  Before you enable Db2 data sharing ......................................................................................... 653
  Application design planning for data sharing ............................................................................. 656
Strategy for creating a Db2 data sharing group ................................................................................ 661
Subsystem parameters and application defaults in a data sharing environment ................................ 663
  Specifying a member's subsystem parameter load module ................................................... 663
  The scope and uniqueness of Db2 subsystem parameters in data sharing ........................... 663
  Updating subsystem parameters for a member ........................................................................ 666
  Application defaults parameters ............................................................................................... 667
Installing a new data sharing group ............................................................................................... 667
Enabling Db2 data sharing .............................................................................................................. 668
  Running the CLIST for enabling data sharing ........................................................................ 669
  Stopping Db2 activity ................................................................................................................. 670
Modify the Db2 definition to IRLM and MVS for data sharing: DSNTIJMV .................................... 671
Update address space startup procedures for Db2 data sharing: DSNTIJMA ................................. 673
Altering the current active log data sets: DSNTIJIN ................................................................. 674
Defining Db2 initialization parameters for enabling data sharing ............................................... 674
Recording Db2 data to SMF (optional) .......................................................................................... 676
Connecting Db2 to TSO .................................................................................................................. 677
Defining Db2 to z/OS for enabling data sharing ............................................................................ 678
Starting the Db2 subsystem .................................................. 678
Creating the default storage group and defining temporary work files: DSNTIJTM .................................................. 680
Creating an image copy of the Db2 directory and catalog: DSNTIJIC (optional) .................................................. 681
Verifying that enabling data sharing was successful (optional) .................................................. 682
Adding a new data sharing member .................................................. 682
Running the CLIST for adding a new member .................................................. 682
Define the new Db2 data sharing member to IRLM and MVS: DSNTIJMV .................................................. 683
Define address space startup procedures for the new Db2 data sharing member: DSNTIJMA .................................................. 685
Defining system data sets: DSNTIJIN .................................................. 686
Defining Db2 initialization parameters for new data sharing members .................................................. 687
Initializing system data sets: DSNTIJID .................................................. 688
Defining user authorization exit routines: DSNTIJEX (optional) .................................................. 688
Recording Db2 data to SMF (optional) .................................................. 688
Establishing subsystem security: DSNTIJRA (optional) .................................................. 690
Connecting Db2 to IMS (optional) .................................................. 690
Connecting CICS to Db2 (optional) .................................................. 690
Defining Db2 to z/OS for adding data sharing members .................................................. 691
Starting the Db2 subsystem .................................................. 691
Creating the default storage group, defining temporary work files, and binding Db2 REXX Language Support: DSNTIJTM .................................................. 693
Creating an image copy of the Db2 directory and catalog: DSNTIJIC .................................................. 694
Verifying that adding a data sharing member was successful .................................................. 694
Ensuring that the WLM address space is defined and available .................................................. 695
Merging existing Db2 data into the group .................................................. 695
Merging subsystems .................................................. 695
Merging data .................................................. 696
Release coexistence in Db2 data sharing .................................................. 703
Special considerations for WLM-managed stored procedures in coexistence .................................................. 704
Determining the release of a data sharing group in coexistence .................................................. 705
Determining the function level of an IRLM group in coexistence .................................................. 705
Call attachment and TSO attachment in coexistence .................................................. 707
Automatic binds in coexistence .................................................. 707
Recommendations for BIND in coexistence .................................................. 708
Recommendations for utilities in coexistence .................................................. 709
Recommendation for group restart in coexistence .................................................. 710
Recommendation for SPUFI in coexistence .................................................. 710
IRLM service and release coexistence .................................................. 710
Preparing for Db2 12 migration in Db2 data sharing .................................................. 711
Migrating an existing data sharing group to Db2 12 .................................................. 712
Migrating the first member of a group to Db2 12 .................................................. 712
Migrating subsequent members of a group to Db2 12 .................................................. 714
Falling back and remigrating in a data sharing environment .................................................. 722
Falling back in a data sharing environment .................................................. 722
Remigrating in a data sharing environment .................................................. 723
Activating Db2 function levels in a data sharing environment .................................................. 724
Starting inactive data sharing group members in a new Db2 release .................................................. 724
Job DSNTIJCB .................................................. 725
Testing the data sharing group .................................................. 726
Testing group buffer pool caching .................................................. 726
Testing global lock serialization .................................................. 727
Testing concurrency .................................................. 727
Disabling and re-enabling Db2 data sharing .................................................. 727
Disabling data sharing .................................................. 728
Data recovery after disabling Db2 data sharing .................................................. 731
Re-enabling Db2 data sharing .................................................. 731

Chapter 10. Verifying successful Db2 12 installation or migration with sample applications .................................................. 733
Installation verification phases and programs .................................................. 734
Planning for verification .................................................. 738
COBOL options to use with the installation verification procedures .................................................. 739
C and C++ language options to use with the installation verification procedures ........................................ 740
PL/I options to use with the installation verification procedures ................................................................. 741
Phase 0: Deleting the sample objects (DSNTEJ0) ......................................................................................... 741
Phase 1: Creating and loading sample tables
  Job DSNTEJ1 ........................................................................................................................................ 743
  Job DSNTEJ1L ....................................................................................................................................... 743
  Job DSNTEJ1P ....................................................................................................................................... 746
  Job DSNTEJ1U ....................................................................................................................................... 747
Phase 2: Testing the batch environment.
  Job DSNTEJ2A ....................................................................................................................................... 748
  Job DSNTEJ2C ....................................................................................................................................... 749
  Job DSNTEJ2D ....................................................................................................................................... 749
  Job DSNTEJ2E ....................................................................................................................................... 750
  Job DSNTEJ2F ....................................................................................................................................... 751
  Job DSNTEJ2H ....................................................................................................................................... 751
  Job DSNTEJ2P ....................................................................................................................................... 752
  Job DSNTEJ2U ....................................................................................................................................... 753
Phase 3: Testing SPUI, DRDA access, dynamic SQL, and TSO
  Testing SPUI .......................................................................................................................................... 755
  Dynamic SQL and the ISPF/CAF application.
  Jobs DSNTEJ3C and DSNTEJ3P .............................................................................................................. 757
  Job DSNTEJ3M ....................................................................................................................................... 759
  Starting an application in an ISPF/TSO environment ............................................................................... 759
Phase 4: Testing the IMS environment
  Jobs DSNTEJ4C and DSNTEJ4P .............................................................................................................. 760
  Starting an application in an IMS environment ......................................................................................... 763
  Using the phone application in IMS ........................................................................................................ 764
Phase 5: Testing the CICS environment
  Job DSNTEJ5A ....................................................................................................................................... 765
  Jobs DSNTEJ5C and DSNTEJ5P .............................................................................................................. 765
  Starting an application in a CICS environment ......................................................................................... 768
  Using the phone application in CICS ........................................................................................................ 769
  Using CICS storage-handling facilities .................................................................................................... 769
Phase 6: Accessing data at a remote site
  DRDA access sample ............................................................................................................................... 771
  Job DSNTEJ6 ....................................................................................................................................... 772
  Starting an application in an ISPF/TSO environment in phase 6 ............................................................ 773
  Stored procedure samples ......................................................................................................................... 773
Phase 7: Accessing LOB data
  Job DSNTEJ7 ....................................................................................................................................... 792
  Job DSNTEJ71 (optional) ......................................................................................................................... 793
  Job DSNTEJ73 ....................................................................................................................................... 794
  Job DSNTEJ75 ....................................................................................................................................... 794
  Job DSNTEJ76 ....................................................................................................................................... 795
  Job DSNTEJ77 ....................................................................................................................................... 796
  Job DSNTEJ78 ....................................................................................................................................... 796
  Starting an application in an ISPF/TSO environment in phase 7 ............................................................ 797

The sample applications
  Printing options for the sample application listings .................................................................................... 798
  The sample application panels .................................................................................................................. 798

Working with scenarios
  Working with the project application scenario ........................................................................................... 801
  Working with the organization application scenario .................................................................................. 804
  Working with the phone application scenario. .......................................................................................... 808
  Working with the distributed organization application scenario. ............................................................ 812
  Working with the employee resume and photo scenarios ........................................................................ 821

Edit exit routine ........................................................................................................................................ 824

Huffman compression exit routine ............................................................................................................. 825

Sample field procedure ............................................................................................................................... 825

Dynamic SQL statements: DSNTESA, DSNTESQ
DSNTESA ................................................................................................................................................ 825
Chapter 11. Working with additional capabilities for Db2 ............................................. 829
Installation information for procedures and functions that are supplied with Db2 ........... 829
  WLM management of stored procedures ............................................................... 835
  Core WLM environments for Db2-supplied routines ........................................... 836
  DSNWLM TOOL .................................................................................................. 843
Sample SQL statements for installing Db2-supplied routines ........................................... 846
Configuration control statements for Db2-supplied routines ........................................... 846
Disabling Db2-supplied routines for validation ............................................................ 848
Setting up support for external SQL procedures ......................................................... 848
Additional steps for enabling the stored procedures and objects for XML schema support. .................................................................................................................. 849
Additional steps for enabling the function for XSLTRANSFORM routines support .......... 850
Additional steps for enabling the administrative task scheduler and administrative enablement routines ............ 851
Additional steps for enabling the CICS transaction invocation procedure ................. 854
Additional steps for enabling the common SQL API stored procedures ....................... 854
Additional steps for enabling WebSphere MQ user-defined functions ......................... 856
Enabling Db2 web services ............................................................................................ 860
Creating REST user-defined functions ............................................................................ 865
Sample SQL procedures for tracing native SQL routines ............................................... 868
Sample IBM InfoSphere BigInsights user-defined functions ....................................... 869
  Creating sample IBM BigInsights user-defined functions ........................................ 869
  Job DSNTEJBI ............................................................................................................... 869
The IBM Data Server Driver for JDBC and SQLJ ......................................................... 871
Connecting the IMS attachment facility ........................................................................ 871
  Making Db2 load modules available to IMS .............................................................. 872
  Defining Db2 to IMS .................................................................................................... 872
  IMS attachment facility macro (DSNMAPN) .............................................................. 878
Connecting distributed database systems ...................................................................... 879
  The communications protocols ................................................................................. 879
  The role of the communications database (CDB) ....................................................... 880
  The DRDA database protocol ..................................................................................... 881
  DDF installation and implications for Db2 ................................................................ 881
Connecting systems with VTAM .................................................................................... 882
  Customize VTAM for Db2 ........................................................................................... 884
  Choosing names and a password ................................................................................ 884
  Defining the Db2 subsystem to VTAM ....................................................................... 887
  Populating the communications database .................................................................. 894
  Starting VTAM to use Db2 ......................................................................................... 898
  Tuning the VTAM system ......................................................................................... 898
  Sample VTAM definitions to connect two Db2 subsystems ....................................... 910
  Using the change log inventory utility to update the BSDS ...................................... 916
Connecting systems with TCP/IP .................................................................................. 917
  TCP/IP terminology .................................................................................................. 918
  Enabling TCP/IP communication .............................................................................. 919
  TCP/IP limitations ...................................................................................................... 931
  Initializing a TCP stack for use with a VIPA ............................................................ 932
  Initializing a TCP stack for use with both IPv4 and IPv6 addresses ....................... 932
  Two-phase commit for TCP/IP clients ....................................................................... 933
  Multiple TCP/IP stacks and Db2 ............................................................................... 934
  Specifying security requirements ............................................................................. 934
Setting up the Db2 command line processor ................................................................. 935
IBM Text Search for Db2 for z/OS .................................................................................. 936
Creating database objects that support query acceleration ............................................ 936
Migrating query acceleration tables to universal table spaces .................................... 937

Chapter 12. Working with international data ................................................................. 941
Character conversion concepts ...................................................................................... 941
System-coded character set identifiers ........................................................................ 942
### Information resources for Db2 12 for z/OS and related products

- Unicode support in Db2 .................................................. 942
- Customizing support for Unicode ...................................... 942
- EBCDIC and ASCII support ............................................. 947
- Implications of character conversions ................................. 950
- Euro symbol support ..................................................... 950
- How an entry in SYSIBM.SYSTRINGS works with character conversion ......................................................... 952
- Conversion-related changes that require remote packages to be rebound ......................................................... 954
- Locales for uppercase and lowercase conversion of EBCDIC data ......................................................... 954
- Setting up z/OS to allow uppercase and lowercase conversion of Unicode and ASCII data ......................................... 955

### Notices

- Programming interface information .................................. 960
- Trademarks ......................................................................... 961
- Terms and conditions for product documentation ................. 962
- Privacy policy considerations ........................................... 962

### Glossary

- ................................................................. 965

### Index

- ................................................................. 967
About this information

This information describes installation and migration of Db2® for z/OS® (Db2) in a data sharing and non-data-sharing environment.

Throughout this information, “Db2” means “Db2 12 for z/OS”. References to other Db2 products use complete names or specific abbreviations.

Important: To find the most up to date content, always use IBM® Knowledge Center which is continually updated as soon as changes are ready. PDF manuals are updated only when new editions are published, on an infrequent basis.

Overview of Db2 12 new function availability
The availability of new function depends on the type of enhancement, the activated function level, and the application compatibility levels of applications. In the initial Db2 12 release, most new capabilities are enabled only after the activation of function level 500 or higher.

Virtual storage enhancements
Virtual storage enhancements become available at the activation of the function level that introduces them or higher. Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release. That is, activation of function level 500 introduces no virtual storage enhancements.

Subsystem parameters
New subsystem parameter settings are in effect only when the function level that introduced them or a higher function level is activated. Most subsystem parameter changes in the initial Db2 12 release take effect in function level 500. For a list of these changes, see Subsystem parameter changes in Db2 12 (Db2 for z/OS What’s New?).

Optimization enhancements
Optimization enhancements become available after the activation of the function level that introduces them or higher, and full prepare of the SQL statements. When a full prepare occurs depends on the statement type:
• For static SQL statements, after bind or rebind of the package
• For non-stabilized dynamic SQL statements, immediately, unless the statement is in the dynamic statement cache
• For stabilized dynamic SQL statements, after invalidation, free, or changed application compatibility level

Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. That is, function level 500 introduces no optimization enhancements.

SQL capabilities
New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous
Db2 releases, including Db2 11 and DB2® 10. For details, see Application compatibility levels in Db2 (Db2 Application programming and SQL).

Who should read this information

This information is primarily intended for those people who are responsible for installing Db2 or setting up Db2 for distributed communications. This information is intended for those people who plan to install Db2 from the enterprise server using the installation CLIST.

This information assumes that you are familiar with:

- The basic concepts and facilities of Db2 in the z/OS environment
- The z/OS Time Sharing Option (TSO) and the z/OS Interactive System Productivity Facility (ISPF)
- The basic concepts of Structured Query Language (SQL)
- The basic concepts of Customer Information Control System (CICS®)
- The basic concepts of Information Management System (IMS)
- How to define and allocate z/OS data sets by using z/OS job control language (JCL)
- How to use IBM System Modification Program/Extended (SMP/E) to install IBM licensed programs

To set up Db2 for distributed communications, knowledge of Virtual Telecommunications Access Method (VTAM®) or Transmission Control Protocol/Internet Protocol (TCP/IP) is also needed.

Db2 Utilities Suite for z/OS

Important: In this Db2 12, Db2 Utilities Suite for z/OS is available as an optional product. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them.

Db2 12 utilities can use the DFSORT program regardless of whether you purchased a license for DFSORT on your system. For more information, see the following informational APARs:

- II14047
- II14213
- II13495

Db2 utilities can use IBM Db2 Sort for z/OS (5655-W42) as an alternative to DFSORT for utility SORT and MERGE functions. Use of Db2 Sort for z/OS requires the purchase of a Db2 Sort for z/OS license. For more information about Db2 Sort for z/OS, see Db2 Sort for z/OS

Related concepts:

- Db2 Utilities Suite for z/OS

Terminology and citations

When referring to a Db2 product other than Db2 for z/OS, this information uses the product's full name to avoid ambiguity.
About the Db2 brand change: IBM is rebranding DB2 to Db2. As such, there will be changes to all the Db2 offerings. For example, “DB2 for z/OS” is now referred to as “Db2 for z/OS,” beginning with Db2 11. While IBM implements the change across the Db2 family of products, you might see references to the original name “DB2 for z/OS” or “DB2” in different IBM web pages and documents. “DB2 for z/OS” and “Db2 for z/OS” refer to the same product, when the PID, Entitlement Entity, version, modification, and release information match. For more information, see [Revised naming for IBM Db2 family products](#).

The following terms are used as indicated:

**Db2** Represents either the Db2 licensed program or a particular Db2 subsystem.

**Tivoli® OMEGAMON® XE for Db2 Performance Expert on z/OS**

Refers to any of the following products:

- IBM Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
- IBM Db2 Performance Monitor on z/OS
- IBM Db2 Performance Expert for Multiplatforms and Workgroups
- IBM Db2 Buffer Pool Analyzer for z/OS

**C, C++, and C language**

Represent the C or C++ programming language.

**CICS** Represents CICS Transaction Server for z/OS.

**IMS** Represents the IMS Database Manager or IMS Transaction Manager.

**MVS™** Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

**RACF®**

Represents the functions that are provided by the RACF component of the z/OS Security Server.

---

**Accessibility features for Db2 12 for z/OS**

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

**Accessibility features**

The following list includes the major accessibility features in z/OS products, including Db2 12 for z/OS. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size

**Tip:** The IBM Knowledge Center (which includes information for Db2 for z/OS) and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

**Keyboard navigation**

For information about navigating the Db2 for z/OS ISPF panels using TSO/E or ISPF, refer to the z/OS TSO/E Primer, the z/OS TSO/E User’s Guide, and the z/OS ISPF User’s Guide. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.
Related accessibility information

IBM and accessibility

See the IBM Accessibility Center at [http://www.ibm.com/able](http://www.ibm.com/able) for more information about the commitment that IBM has to accessibility.

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other Db2 for z/OS documentation.

Send your comments by email to [db2zinfo@us.ibm.com](mailto:db2zinfo@us.ibm.com) and include the name of the product, the version number of the product, and the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title or a help topic title).

How to read syntax diagrams

Certain conventions apply to the syntax diagrams that are used in IBM documentation.

Apply the following rules when reading the syntax diagrams that are used in Db2 for z/OS documentation:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.
  - The ►► symbol indicates the beginning of a statement.
  - The ───► symbol indicates that the statement syntax is continued on the next line.
  - The ◄── symbol indicates that a statement is continued from the previous line.
  - The ───◄ symbol indicates the end of a statement.
- Required items appear on the horizontal line (the main path).

  ►►required_item───────►

- Optional items appear below the main path.

  ►►required_item───►optional_item───►

  If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

  ───►optional_item───────►

- If you can choose from two or more items, they appear vertically, in a stack.
  If you must choose one of the items, one item of the stack appears on the main path.

  ►►required_item───►required_choice1─►required_choice2

If choosing one of the items is optional, the entire stack appears below the main path.

►►**required_item**

  - **optional_choice1**
  - **optional_choice2**

If one of the items is the default, it appears above the main path and the remaining choices are shown below.

►►**required_item**

  - **default_choice**
  - **optional_choice**
  - **optional_choice**

- An arrow returning to the left, above the main line, indicates an item that can be repeated.

►►**required_item**

  - **repeatable_item**

If the repeat arrow contains a comma, you must separate repeated items with a comma.

►►**required_item**

  - **repeatable_item**

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.

►►**required_item**

  - **fragment-name**

*fragment-name:*

►►**required_item**

  - **optional_name**

- With the exception of XPath keywords, keywords appear in uppercase (for example, `FROM`). Keywords must be spelled exactly as shown. XPath keywords are defined as lowercase names, and must be spelled exactly as shown. Variables appear in all lowercase letters (for example, `column-name`). They represent user-supplied names or values.

- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

**Related concepts:**

- [Commands in Db2 (Db2 Commands)]
- [Db2 online utilities (Db2 Utilities)]
Chapter 1. Installing and migrating Db2

The essential steps are similar for both installing Db2 12 and migrating to Db2 12. When you migrate to Db2 12, you must also activate function level 500 or higher before most new capabilities can be used.

Before you begin

Prepare your system for Db2 installation or migration, as described in Chapter 2, “Preparing your system to install or migrate to Db2 12,” on page 5.

About this task

Installation and migration tools

Db2 provides the following tools that help you complete the process of installing or migrating to Db2 12:

- Most of the job control language (JCL) that is needed to install and migrate Db2 12. This JCL constitutes the installation and migration jobs. Each of these jobs helps you perform an installation or migration task.

- The installation CLIST (command list) to help tailor the installation and migration jobs. This CLIST is also called the migration CLIST, or simply the CLIST. It contains the necessary code for tailoring the jobs to suit your needs.

Tip: You can capture most settings of an existing Db2 subsystem by running the DSNTIJXZ job, which invokes the DSNTXAZP tool. Running this job reduces the effort to manually compare settings in the current DSNTIJUZ job with the values that are defined in the defaults input member. For details, see Updating the CLIST defaults input member: job DSNTIJXZ and DSNTXAZP tool.

- A series of ISPF panels that you can use to pass information to the CLIST. With Interactive Systems Productivity Facility (ISPF) and Interactive Systems Productivity Facility/Program Development Facility (ISPF/PDF), you can use a series of ISPF panels to pass parameter values to the CLIST. The CLIST uses these values to tailor the installation and migration jobs. This process is called the ISPF tailoring session.

- Sample applications to help determine if you installed or migrated Db2 correctly. Db2 provides a set of sample programs and procedures that help you determine if Db2 is operating correctly.

All references to SYS1.PARMLIB also imply the logical PARMLIB data set that is used for Db2.

Minimal assemblies

Because it is distributed as object code, Db2 requires few assemblies. You must complete an assembly to specify Db2 initialization parameters, but this assembly requires only a few seconds.

Ability to defer decisions about Db2 characteristics:

Db2 allows you to specify many subsystem characteristics during Db2 operation. You can modify initialization parameters, authorize users, define databases and tables, and tune Db2 while Db2 is running. Therefore, you can defer many decisions until after you finish installing or migrating Db2.
Ability to update installation and migration options

During the process of installing and migrating, Db2 uses ISPF panels to prompt you for many options. Db2 allows you to update most of these options without requiring you to reinstall or remigrate. You can accept the default values for certain options and, after acquiring experience with Db2, tailor them to suit your needs.

Procedure

To install or migrate to Db2 12, you complete the following procedures:

1. Estimate storage needs.
2. Determine which new functions you need.
3. If you are using distributed data, install VTAM and, optionally, TCP/IP.
4. Set up a Parallel Sysplex® if you plan to use data sharing.
5. Load the Db2 libraries (do the SMP/E steps). If you plan to use the callable SQL interface of Db2, see Configuring Db2 ODBC and running sample applications (Db2 Programming for ODBC) for the additional installation jobs that you need to run. If you plan to use Db2 for z/OS Java™ Edition, see Installing the IBM Data Server Driver for JDBC and SQLJ for additional installation jobs that you need to run.
6. If you are migrating, install needed service on the prior release. See the Db2 Program Directory for information about needed service.

Important: Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

7. If you are migrating, check release incompatibilities, and make the necessary changes in your applications.
8. If you are installing, provide an SMS environment for managing Db2 catalog and directory data sets. The SMS environment must include a data class for allocating the data sets in extended format and using extended addressability. You are not required to convert existing Db2 catalog and directory data sets to the SMS environment before migrating to Db2 12. These data sets will be converted the next time the related table space is reorganized.
9. Tailor the installation or migration jobs.

Tip: Consider using z/OSMF workflows to automate the remainder of the installation or migration process. For more information, see Chapter 7, “Automating Db2 installation or migration with z/OS Management Facility,” on page 563

10. Install or migrate to Db2 12.
11. Connect the Db2 attachment facilities.
12. Prepare Db2 for use.
13. Verify the Db2 12 installation or migration to Db2 12.
14. Verify the Db2 12 installation or migration
15. If you encounter problems during or after migration to Db2 12 before the activation of function level 500 or higher, you can complete the following procedures:
   a. Fall back to Db2 11.
b. Remigrate to Db2 12.

16. Issue an ACTIVATE command to activate function level 500 or higher. The use of most new capabilities that Db2 12 introduces is prevented until you complete this step. For detailed information about activating function level 500 or higher, see “Migration step 29: Activate function level 500 or higher” on page 538. For an overview of continuous delivery and function levels in Db2 12, see “What’s new in Db2 12 function levels (Db2 for z/OS What’s New?)”.

**Important:** Do not issue the ACTIVATE command or run job DSNTIJAF for activation of function level 500 or higher until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

**Related tasks:**
- Chapter 9, “Installing, migrating, and enabling Db2 data sharing,” on page 601
- Installing the RACF access control module (RACF Access Control Module Guide)

**Related reference:**
- z/OS Parallel Sysplex Configuration Overview (IBM Redbooks)

**Related information:**
- IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)
Chapter 2. Preparing your system to install or migrate to Db2 12

The key to a successful installation or migration of Db2 is planning. Prepare your environment and system before you begin installing or migrating.

Before you begin

Familiarize yourself with the various changes that Db2 12 introduces, as described in What’s changed in Db2 12 (Db2 for z/OS What's New?).

In particular, plan to resolve any incompatible changes that might impact your migration to Db2 12. For a summary of such changes, see Incompatible changes in Db2 12 (Db2 for z/OS What’s New?) and Function that Db2 12 no longer supports (Db2 for z/OS What’s New?).

About this task

Plan the amount of direct-access storage and virtual storage that you need. Planning and coordinating with other Db2 subsystems is essential if you plan to install the distributed data facility (DDF). Review the values that are needed for the parameters on the installation and migration panels. By planning in advance, your task of filling in the parameters becomes easier.

Data Sharing

For information about specific planning for Db2 data sharing, see Planning for Db2 data sharing.

Related tasks:
- Tailoring Db2 12 installation and migration jobs with the CLIST
- Controlling connections to remote systems (Db2 Administration Guide)

Hardware and software requirements for Db2

Before you can install or migrate Db2 12, you must fulfill the hardware and software requirements.

You must migrate to a z/OS Version 2 Release 1 or later environment before installing or migrating to Db2 12. Db2 12 for z/OS operates on IBM System z114 or z196 and subsequent 64-bit z/Architecture® processors.

See the Db2 Program Directory for information about system requirements.

Related reference:
- Summary of operational requisite changes for Db2 function levels (Db2 for z/OS What's New?)
Required maintenance for Db2 12 installation or migration

In addition to ensuring that you have required hardware and software products installed, you must ensure that required service changes for Db2 12 have been applied.

To help you identify required service, z/OS provides enhanced HOLDDATA. Enhanced HOLDDATA categories (FIXCATs) provide a simplified, automated method of identifying and applying missing PTFs that are required for installation or migration or to support specific product functions. For installation and migration to Db2 12, use the following FIXCATs:

- IBM.Migrate-Fallback.DB2.V12
- IBM.Coexistence.DB2.SYSPLEXDataSharing

You can use this automated process instead of manually reviewing Preventive Service Planning (PSP) buckets, identifying service that is already installed, and applying missing PTFs.

For more information about enhanced HOLDDATA and available FIXCATs, see [Enhanced HOLDDATA for z/OS](#).

Related concepts:

- Program directories for Db2 12

Related information:

- [Getting an RSU](#)
- [IBM z/OS Consolidated Service Test and the RSU](#)

Required authorization for installation or migration

You are not required to use the SYSADM or installation SYSADM authority to install or migrate a Db2 subsystem. Instead, you can use the installation SYSOPR authority to perform all installation or migration tasks while preventing unintended access to user data.

In Db2 11 and previous releases, you must have the SYSADM authority to perform Db2 installation or migration tasks. In addition, you are required to have the installation SYSADM authority to run the DSNTIJTC job and the CATMAINT utility. However, the SYSADM and installation SYSADM authorities provide unrestricted access to all user objects in the subsystem, which in some cases is unintended and introduces unnecessary security risks.

Using the installation SYSOPR authority can help reduce those risks. The installation SYSOPR authority provides all the privileges that are required for installing or migrating Db2. For example, with installation SYSOPR, you can access catalog and directory objects, create, drop, or alter system objects, execute the -ACTIVATE command, specify the owner for the BIND and REBIND commands, and run the CATMAINT utility. However, you are not allowed any access to non-system objects or user data in the subsystem.

The overall process of installing or migrating Db2 with the installation SYSOPR authority is the same as that with the SYSADM authority. You can use the installation SYSOPR authority to perform all installation or migration tasks if you meet the following additional requirements:
The primary or secondary authorization ID that performs installation or migration tasks must have the installation SYSOPR authority.

The current SQLID that processes data definition statements must be set to SYSINSTL which owns all the system objects created by the installation SYSOPR authority.

The OWNER option for the BIND and REBIND commands must be set to an ID that is authorized to bind and execute SQL statements in a package. To obtain the required authorization for migration to Db2 12, consider granting system DBADM authority with DATAACCESS to an ID that is authorized to bind and execute SQL statements. Revoke DBADM authority with DATAACCESS from that ID after migration is complete.

For installing Db2 12, specify an installation system administrator ID.

The DSNTIAD plan must be used with the DSNTIAD program to process all dynamic data definition statements for Db2 installation or migration. The DSNTIAD plan is specifically provided for dynamically executing SQL statements when the primary or secondary authorization ID is granted the installation SYSOPR authority and the current SQLID is set to SYSINSTL. The DSNTIAD plan is bound by job DSNTIJTM when it is configured by the CLIST for installation or migration.

An explicit qualifier must not be specified for the CREATE TABLE or CREATE INDEX statement. The qualifier can be implicitly specified through a leading SET CURRENT SCHEMA statement so that SYSINSTL can own the table or index.

You can configure installation or migration jobs to use the installation SYSOPR authority by running the CLIST and specifying SYSINSTL for the ROUTINES CREATOR, SEC DEF CREATOR, and INSTALL SQL ID fields on panel DSNTIPG. When INSTALL SQL ID is set to SYSINSTL, the authorization ID with the installation SYSOPR authority can create, alter, or drop the following objects:

- Auxiliary table
- Created global temporary table
- Database
- Function
- Index
- Procedure
- Procedure and function in schema, SYSTOOLS and SYSFUN
- Storage group
- Table
- Table space
- Trigger

You must explicitly specify the owner of a procedure or function package. If a stored procedures is created with the SECURITY DEFINER clause, you must define SYSINSTL in RACF. When the current SQLID is set to SYSINSTL, the authorization ID with the installation SYSOPR authority can grant certain privileges on some system objects and resources, as follows:

- All database, table space, and table privileges on objects in database DSNDB04 and system databases DSNRFDB, DSNRLST, DSNOPTDB, DSNMDCDB, DSNADMDB, DSNATPDB, DSN5JSDB, DSNMQDB, SYSIBMTA, SYSIBMITS, and DSNXSR
- The USE privilege on buffer pools and storage groups
• All privileges on plans that begin with 'DSN'
• All privileges on packages where the collection-ID and package-name begin with 'DSN'
• The EXECUTE privilege on system-defined routines.

In addition, the same authorization ID can create, alter, and drop the following objects without any additional privileges:
• Alias
• Distinct type
• Distinct type in schema, SYSTOOLS and SYSFUN
• Sequence
• Variable.

With additional privileges, the authorization ID can create and alter the following objects:
• A function that has any of the following characteristics: it uses a table as a parameter, it is sourced on another function, or it is defined with LANGUAGE JAVA and the EXTERNAL NAME clause specifies the name of a JAR file.
• An index that uses an expression.
• A procedure that is defined with LANGUAGE is JAVA and the EXTERNAL NAME clause specifies the name of a JAR file.
• A trigger when additional privileges are granted for executing SQL in the trigger definition.
• A view when the SELECT privilege on the base table or view is granted.

Related reference:
DSNTIPG: Installation preferences panel
DSNTIPP1: Protection panel 2

Storage requirements for Db2

When you install or migrate Db2, consider the size of your site and how it will be set up to determine your storage needs.

You can use the installation CLIST to calculate your estimated storage needs. For details, see Calculating storage requirements with the installation CLIST.

The parameters that you specify when you run the installation CLIST affect the sizes of some data sets and the required amount of virtual storage. All data sets are linear data sets, with the exception of the bootstrap data set, which is a key-sequenced data set.

You can use Data Facility Storage Management Subsystem (DFSMS) to manage Db2 data sets. It provides automatic backup and recovery features, which might require disk storage beyond what is estimated below. Starting in DB2 10, data sets for the Db2 catalog and directory are managed by Db2 and must be SMS-managed. The SMS environment that you used for DB2 10 can be used for Db2 12. You are not required to convert existing Db2 catalog and directory data sets to the SMS environment before migrating to Db2 12. These data sets will be converted the next time the related table space is reorganized.

Related concepts:
Storage estimates for data sharing environments
Calculating storage requirements with the installation CLIST

By providing information about the sizes of your Db2 environment, you can use the installation CLIST to calculate certain storage requirements for your Db2 subsystem.

Before you begin

For information about increased EDM storage requirements in Db2 12, see Increased EDM storage usage for DBDs (in Storage release incompatibilities).

About this task

The CLIST contains the algorithms that Db2 uses to calculate storage based on the parameters that you supply during installation or migration.

Procedure

1. Run the installation CLIST, as described in Tailoring Db2 12 installation and migration jobs with the CLIST.
2. In the INSTALL TYPE field on panel DSNTIPA1, specify INSTALL. If you specify MIGRATE for the INSTALL TYPE field, Db2 12 uses values from Db2 11 instead of running the CLIST calculations.
3. On panel DSNTIPD, specify field values that describe the size of your Db2 environment. Specify integer values, not fractions. You can use K (as in 32 K) for multiples of 1024 bytes and M (as in 16 M) for multiples of 1,048,576 bytes in most fields. do not exceed the maximum value that is accepted by the field. The CLIST uses the values that you specify to calculate storage requirements. You can see the calculation results on the fields of the CLIST calculations panel: DSNTIPC.
4. On panel DSNTIPC, specify whether to accept the calculated values.
   The Override column of panel DSNTIPC controls whether Db2 uses the calculated value:
   • To accept the calculated value, ensure that the Override column contains no value.
   • To override the calculated value, specify a value in the Override column.

What to do next

If you want to examine the algorithms that Db2 uses for the CLIST calculations, that Db2 uses, print or edit the installation CLIST and REXX EXEC. The DSNTCALC EXEC contains most of the data set calculations.

Related concepts:
- Storage release incompatibilities (Db2 for z/OS What's New?)

Related reference:
- DSNTIPD: Sizes panel 1
Db2 subsystem storage requirements

You must configure storage sizes for your Db2 subsystem.

You can use the installation CLIST to calculate storage requirements for your Db2 installation. For details, see Calculating storage requirements with the installation CLIST.

If you do not accept the default values, you can calculate the storage that you need for the Db2 data sets by using the information in **Active log data sets storage requirements**. For other data sets, you can use the formulas in the CLIST. After calculating the required storage for each data set, you can calculate the total requirements.

Although the Db2 libraries require a fixed amount of space, disk requirements for active logs and the Db2 catalog increase with the size of a site. You need additional space for archive logs, image copies, user databases, and other working data sets.

**Related reference:**
- [DSNTIPD: Sizes panel 1](#)

**Related information:**
- [Db2 for z/OS Program Directories](#)

Db2 catalog storage requirements

Estimated storage requirements for the Db2 catalog include space for the entire set of Db2 catalog data sets and their indexes.

Data sets for the Db2 catalog and directory are managed by Db2 and must be SMS-managed. The SMS environment that you use for Db2 catalog and directory data sets must be established before you begin installation or migration to Db2 12. The SMS environment must include a data class for allocating the data sets in extended format and using extended addressability. You are not required to convert existing Db2 catalog and directory data sets to the SMS environment before migrating to Db2 12. These data sets will be converted the next time that the related table space is reorganized.

**Related tasks:**
- [Changing catalog and directory size and location (Db2 Performance)](#)

**Related reference:**
- [DSNTIPC: CLIST calculations panel 1](#)

Db2 directory storage requirements

The amount of directory space required for the Db2 directory depends mainly on the number of user databases, application plans and packages, and tables in the Db2 subsystem.

Data sets for the Db2 catalog and directory are managed by Db2 and must be SMS-managed. The SMS environment that you use for Db2 catalog and directory data sets must be established before you begin installation or migration to Db2 12. The SMS environment must include a data class for allocating the data sets in extended format and using extended addressability. You are not required to
convert existing Db2 catalog and directory data sets to the SMS environment
before migrating to Db2 12. These data sets will be converted the next time the
related table space is reorganized.

A typical site requires about 808 cylinders for the Db2 directory.

Related reference:
[DSNTIPC: CLIST calculations panel 1]

Active log data sets storage requirements
Active log data sets record significant events and data changes. Active log data sets
are periodically offloaded to archive log data sets. Storage requirements for active
log data sets depend on the frequency of data updates and of offloads to archive
log data sets.

Before function level 500 or higher is activated, the maximum data set size is 4 GB.
If you use 3390 volumes, in the DEFINE CLUSTER command, define the data set
with a primary quantity that is big enough to hold the entire active log, and a
secondary quantity of 0. Ensure that the allocation for each active log data set does
not exceed the following values:

For 3390-9 volumes:
• 87375 tracks, if allocated in units of tracks
• 5825 cylinders, if allocated in units of cylinders

For 3390-A volumes:
• 5817 cylinders. 3390-A volumes are extended address volumes (EAV), for which
  the space allocation unit can be 21 cylinders, instead of one cylinder.

Related information:
[DEFINE CLUSTER command (DFSMS Access Method Services for Catalogs)]

After function level 500 or higher is activated, the maximum data set size is 768
GB. In most cases, the best data set size is considerably less than 768 GB. When
you define a data set that is greater than 4 GB, you need to specify Extended
Format (EF) in the SMS data class definition.

You must also ensure that each archive log data set is at least as large as your
active log data sets. In Db2 12, subsystem parameter ALCUNIT is removed, and
the allocation unit for the PRIQTY and SECQTY subsystem parameters is always
cylinders, so you might need to adjust those values. Set subsystem parameters
PRIQTY and SECQTY so that the archive log can extend to contain the active log.
If the active log data set size is greater than 4 GB, you need to specify Extended
Addressability in the SMS data class definition for the archive log data sets. Doing
this allows a maximum number of 123 extents for each volume, instead of 16
extents for a data set type of BASIC or LARGE. If your SMS data class attribute
Override Space is set to YES, the data class space settings can override the Db2
subsystem parameter settings. In that case, you need to ensure the space quantities
that you specify for the data class allow the archive logs to contain the active logs.

If the archive logs are written to tape, a large data set might require two tapes
instead of one. For real tape, space might also be wasted on the tapes. Therefore,
you must adjust your data set sizes that are based on the size of your real or
virtual tapes. Most tape products use compaction. You can also use DFSMS to
compress the archive log data sets.
When Db2 fills up one active log data set, it must offload the log data to an archive log data set. At the same time, Db2 begins writing to the next active log data set. When Db2 reaches the last active log data set, it starts again with the first active log data set. However, if the applications catch up to the offload process, the applications are suspended until the offload process can finish writing the next active log data set. Such application suspensions often happen because of the following situations:

- Large batch jobs that insert, update, or delete data.
- Frequent inserts.
- Logging of large LOBs.
- The offload operation is too slow.

To prevent such application suspensions, you can use any of the following approaches:

- Enable fast offload operations. For example, you might place the archive log data sets on a disk device that has fast channels. (You might move the data sets to tape later.)
- Allocate a large amount of active log space to support sustained bursts of insert, update, delete activity. After function level 500 or higher is activated in Db2 12, Db2 supports as many as 93 active log data sets with a maximum size of 768 GB each. This limit provides up to 71,424 GB of active log space. You must double the space calculation for dual logging. In many cases, fewer than the maximum number of data sets can be used. Two data sets might be sufficient. However, the more log space that you allocate, the larger a burst of insert, update, and delete operations can be without causing performance problems.

During startup, Db2 allocates the BSDS data set, all of the active log data sets, and, when dual logging is used, all secondary log data sets. When in a data sharing group, Db2 also allocates peer BSDS data sets and active log data sets as necessary during execution. To avoid exhausting the space in the MVS task I/O table (TIOT) for the Db2 master address space, which can cause Db2 to fail, perform the following actions:

1. Set the MVS TIOT size to 64K in the ALLOCxx PARMLIB member.
2. Specify NON_VSAM_XTIOT=YES in the DEVSUPxx PARMLIB member to enable Db2 to use the MVS extended task I/O table (XTIOT) when allocating offload data sets.
3. Specify a dynamic volume count (DVC) of 0 for any SMS data set class that is used to allocate a Db2 BSDS, active log, or secondary log data set.
4. Reallocate any existing BSDS, active log, or secondary log data sets that were allocated with a DVC greater than 0.

Also, as a rule, reading active log data sets is faster than reading archive log data sets, even if the archive log data sets are stored on disk. (A possible exception to this rule is if you use DFSMS compression for the archive logs.) If your recovery performance depends on the speed of reading the logs, then recovery performance can benefit from having as many active log data sets as possible.

Consider the amount of log data that is written and archived each day. For example, assume you want the ability to restore the database to a point in time 48 hours ago and apply the log records from the past 48 hours with good performance. In that case, you must retain 48 hours of log data on disk. If your applications generate 100 GB of log data per day, you need 200 GB of log space on disk, to apply two days of log records from disk. All 200 GB can be saved in active
log data sets if you allocate enough space. If you do not allocate enough space, then the rest of the log data must be read from archive log data sets.

How you plan to manage database recovery and how you manage your archive log data sets are closely related. If your archive log data sets are on tape (no matter whether they are virtual or real tapes), two recover jobs cannot allocate the same tape at the same time. Contention is worse if multiple data sets are stored on a single tape. For that reason, it is often best if the archive logs are stored on real disk devices instead of tape.

However, you can use the following alternative approaches to reduce the cost of storing the archive logs:

- Use inexpensive nearline drives for the archive logs.
- Store the archive log data sets on tape, and recall the archive log data sets from tape to disk before you submit recovery jobs.
- Tolerate slow recovery.

However, remember as you plan for managing Db2 logs, that it is too late to change your approach when you must recover your Db2 database.

**Checkpoint frequency**

The CHECKPOINT TYPE field on panel DSNTIPL1 specifies whether the frequency of system checkpoints is controlled by number of consecutive log records, number of minutes, or both. The RECORDS/CHECKPOINT field and MINUTES/CHECKPOINT field on panel DSNTIPL1 determine the number of log records between checkpoints or the number of minutes per checkpoint. When you choose a value for the checkpoint frequency, you must consider the trade-offs between the costs of frequent subsystem checkpoints and the time that is needed to restart a Db2 subsystem after a termination without a quiesce. If the checkpoint value is more than 1,000,000 log records, the time that is needed to restart Db2 after a termination without a quiesce can grow to over 15 minutes. Typical values for the checkpoint frequency are in the range of 500,000 - 1,000,000 log records or 2 - 5 minutes.

**Change data capture**

When tables are defined with the DATA CAPTURE CHANGES option, the entire before-image of an updated row is captured on the log. This additional information can represent an increase in log data that is compared to tables that are not defined with the DATA CAPTURE CHANGES option, depending on whether the table contains fixed-length or variable-length rows.

Related concepts:
- Logging environment for data sharing (Db2 Data Sharing Planning and Administration)

Related tasks:
- Improving log capacity (Db2 Performance)

Related reference:
- DSNTIPL1: Checkpoint parameters
Bootstrap data sets storage requirements
Each bootstrap data set (BSDS) requires 15 MB. During installation, Db2 automatically allocates two copies of the BSDS. During migration, Db2 12 adopts the BSDS characteristics that you specified for your previous version.

That is, if you specified two copies of the BSDSs for your previous version, you will have two copies for Db2 12. Keeping two copies of the BSDS is strongly recommended. The total space requirement is about 30 MB for both BSDSs. The BSDSs require about 45 cylinders of 3390 storage.

Related concepts:
- Management of the bootstrap data set (Db2 Administration Guide)

Work file database storage requirements
The work file database is used as storage for processing SQL statements that require working storage.

Work file storage is used for sorting, RID pool overflow to work files, and for temporary tables and common table expressions.

Work file usage varies, depending on the number of operations that require work files, and the number of concurrent users that require work files. You can monitor work file usage by gathering Db2 statistics traces. Use the high water mark value for work file usage from the traces as a guideline for determining the amount of space to allocate for work files.

If you are installing Db2, see Storage requirements for installation, below.

Storage requirements for migration
You might need more storage for the work file database if you have a large amount of data to sort and a large amount of concurrent sort activity. If you are sorting compressed data, allow for the same amount of storage that you would need if the data were not compressed. The maximum amount of storage that you need is enough to satisfy the requirements of the largest combination of concurrent activities that use the work file database. The amount of storage that is required for a sort depends on the following variables:

- Data size
- Sort key size

Important: Db2 12 uses more 32KB work files than previous Db2 releases. For migration to Db2 12, you might need to increase your 32KB work file allocations for work file data sets and work file buffer pools.

You can estimate the total amount of work file space that is needed to perform the sort as follows:

- Let MIN be the operation of selecting the lowest value from a set of values.
- Let FLOOR be the operation of discarding the decimal portion of a real number.
- Let CEILING be the operation of rounding a real number up to the next-highest integer.
- Let Data be the total data length in bytes.
- Let Key be the total length of the sort key.
- Let Row-overhead be the 6-byte record header plus the 2-byte record page map entry.
• Let Prefix be the 16-byte header.
• Let Rows be the total number of rows that are being sorted.
• Let Segsize be the segment size.

Then calculate as follows:

\[
\text{Records per page} = \min(\text{MAXROWS}, \ \text{FLOOR}(4056 / (\text{Data + Key + Row-overhead})))
\]

\[
\text{Total pages} = \text{CEILING}(\text{Rows} / \text{Records per page})
\]

\[
\text{Total segments} = \text{CEILING}(\text{Total pages} / \text{Segsize})
\]

\[
\text{Total pages used} = \text{Total segments} + \text{Segsize}
\]

The number of records per page cannot exceed 255 (the value of MAXROWS).

This result tells you how much storage is needed in the work file database after sort processing. However, if a merge phase was required during sort processing, an additional intermediate copy of the records might exist at any given time. For most subsystems, you can assume that about half of the records that are involved in a sort have two copies. Therefore, a multiplier value of 1.5 is safe. If you want to be conservative, choose 2 for your multiplier value. Therefore, the amount of storage that is used in the work file database during sort processing can vary from 1 to 2 times the storage that is needed after sort processing. The actual storage that is used might also increase if you have little available buffer pool storage.

When a large object (LOB) column is part of a result table, and the result table must be placed in a work file for sorting, the actual LOB column data is not placed in the work file. Therefore, LOB columns do not require large increases in the amount of work file space that Db2 requires. For work file calculations, you can assume 51 bytes of storage per LOB column for the work file.

However, if a large object (LOB) column is defined with the INLINE attribute and the result table must be placed in a work file, the inline LOB data will be placed in the work file. Therefore, work file calculations need to account for the length of the inline LOB.

To determine the number of tracks that are needed, convert the number of pages into bytes, and divide the result by the number of bytes per unit. Let \( r \) be the number of 4096-byte records per track, and let safety_factor be a number from 1.5 to 2.0. For 3390 devices, \( r \) is 12.

\[
\text{Tracks} = \text{CEILING}(\text{Total pages used} / r) \times \text{safety factor}
\]

**Example 1:** Consider a table (TABLE1) that contains 45,327 rows, for which you want to create a nonunique padded index on COL1 CHAR(3) NOT NULL, COL2 CHAR(4), COL3 VARCHAR(20), and COL4 SMALLINT. Determine the amount of temporary storage that Db2 needs to create this index as follows:

- \( \text{Data} = 3 + (4 + 1) + (20 + 1) + (2 + 1) + 4 = 36 \)
- \( \text{Key} = 36 \) (Data plus RID is key for CREATE INDEX)
- \( \text{Rows} = 45,327 \)
- \( \text{Record overhead} = 8 \)
- \( \text{Records per page} = \min(\text{MAXROWS}, \ \text{FLOOR}(4056 / (36 + 36 + 12))) = 48 \)
- \( \text{Total pages} = \text{CEILING}(45,327 / 50) = 907 \)
- \( \text{Segments} = \text{CEILING}(907 / 24) = 38 \)
- \( \text{Total pages used} = \text{CEILING}(38 \times 24) = 912 \)
Example 1 is a data page calculation for storing index keys in the work file database. For this example, 114 tracks of a 3390 storage device are needed. The 2-byte length field of a VARCHAR column is not a part of Data for CREATE INDEX. The RID field is a part of Data, and the Key includes the entire Data portion, including the RID.

Example 2: Consider TABLE1 again and the following SQL query:

```sql
SELECT COL1, COL2, COL3, COL4
FROM TABLE1
ORDER BY COL2, COL3, COL1;
```

This query, which includes an ORDER BY clause, requires a sort. Determine the amount of temporary storage that is required for this table as follows:
- \( \text{Data} = 3 + (4 + 1) + (20 + 2 + 1) + (2 + 1) = 34 \)
- \( \text{Key} = (4 + 1) + (20 + 1) + 3 = 29 \)
- \( \text{Rows} = 45,327 \)
- \( \text{Record overhead} = 8 \)
- \( \text{Segsize} = 24 \)
- \( \text{Records per page} = \text{MIN} (\text{MAXROWS}, \text{FLOOR} (4056 / (34 + 29 + 12))) = 54 \)
- \( \text{Total pages} \) (final result) = \( \text{CEILING} \left( \frac{45,327}{57} \right) = 796 \)
- \( \text{Segments} \) (final result) = \( \text{CEILING} \left( \frac{796}{24} \right) = 34 \)
- \( \text{Total pages used} = \text{CEILING} \left( 34 \times 24 \right) = 816 \)
- \( \text{Tracks} = \text{CEILING} \left( \frac{816}{12} \right) \times 1.5 = 102 \)
- \( \text{Total pages} \) (during processing) = \( \text{CEILING} \left( 1.5 \times 796 \right) = 1194 \)
- \( \text{Segments} \) (during processing) = \( \text{CEILING} \left( 1.5 \times 34 \right) = 51 \)
- \( \text{Total pages used} \) (during processing) = \( \text{CEILING} \left( 51 \times 24 \right) = 1224 \)
- \( \text{Tracks} \) (during processing) = \( \text{CEILING} \left( \frac{1224}{12} \right) \times 1.5 = 153 \)

For this example, which is a table calculation, 102 tracks of a 3390 storage device are needed. The 2-byte length field of a VARCHAR column is a part of Data, and the Key does not include the entire Data portion.

You can use the sort summary trace record, IFCID 0096, to simplify some of the calculations. This record shows the number of records that are sorted, the sort record size (Data + Key), and an indication of whether a merge phase was required for an individual sort request.

Also note that if the overall record length (Data + Key + Prefix) exceeds 100 bytes, Db2 attempts to create the work file in a table space with a 32 KB page size. If the overall record length is 100 bytes or less, Db2 uses a table space with a 4 KB page size.

**Storage requirements for installation**

If you are installing Db2, use the following calculations to determine either PRIQTY (primary space allocation) or DSSIZE and MAXPARTITION values for
creating work file table spaces. During installation, you must create at least one 4 KB work file table space and one 32 KB work file table space.

**For 4 KB work file table spaces:**
- Let total space be the total amount of space in megabytes (from the TEMP 4K SPACE field on panel DSNTIP9).
- Let table spaces be the number of table spaces (from the TEMP 4K TBL SPACES field on panel DSNTIP9).
- Let CEILING be the operation of rounding a real number up to the next-highest integer.
- Let GB per table space be the amount of space per work file table space in gigabytes.

Calculate the following:

\[
GB \text{ per table space} = \text{CEILING} \left( \frac{\text{total space}}{\text{table spaces}} \right) / 1024
\]

If GB per table space < 1 GB, then PRIQTY = CEILING \( \left( \frac{\text{total space}}{\text{table spaces}} \right) \) and DSSIZE = 4 GB and MAXPARTITIONS = 4096

Otherwise, DSSIZE = 4 GB and MAXPARTITIONS = 4096

**For 32 KB work file table spaces:**
- Let total space be the total amount of space in megabytes (from the TEMP 32K SPACE field on panel DSNTIP9).
- Let table spaces be the number of table spaces (from the TEMP 32K TBL SPACES field on panel DSNTIP9).
- Let CEILING be the operation of rounding a real number up to the next-highest integer.
- Let GB per table space be the amount of space per work file table space in gigabytes.

Calculate the following:

\[
GB \text{ per table space} = \text{CEILING} \left( \frac{\text{total space}}{\text{table spaces}} \right) / 1024
\]

If GB per table space < 1 GB, then PRIQTY = CEILING \( \left( \frac{\text{total space}}{\text{table spaces}} \right) \) and DSSIZE = 4 GB

If GB per table space ≤ 16 384 GB, then DSSIZE = 32 GB

Otherwise, DSSIZE = 32 GB and MAXPARTITIONS = 4096

Related concepts:

- Diagnostic traces for attachment facilities (Db2 Administration Guide)

Related reference:

DSNTIPC: CLIST calculations panel 1

**Default database storage requirements**

The size of the default database depends on column lengths, page sizes, and index column lengths. The estimated size of your data, multiplied by 2, usually provides an adequate planning estimate for the default database size.

**Dump data set size storage requirements**

When planning for your Db2 subsystem storage, you must reserve enough space for the dump data sets. The storage requirements depend on factors such as the number of dump data sets you have defined and whether you invoke authorized Language Environment® functions.
**Recommendation:** Use the following guidelines to estimate storage requirements for the dump data sets:

- Have at least two dump data sets.
- Have approximately 1000 cylinders for all dump data sets.
  
  If you invoke authorized Language Environment functions, a different style of dump might be generated, which requires approximately 520 cylinders.
- Have 3.25 MB of space for Db2 volatile summary storage data.

Summary data in dumps is usually enough to diagnose most problems. In addition to summary data, Db2 also requests a SDUMP from the operating system to provide these additional storage areas if enough space is available in the dump data set:

- Db2 system services address space
- Db2 database services address space
- Db2 distributed data facility (DDF) address space
- Allied address space of the failing allied task
- IRLM address space for data sharing environments

Db2 passes the following parameters to the SDUMP service aid through the SDATA keyword: SQA, ALLPSA, LSQA, SUMDUMP, and CSA (subpools 231 and 241).

After Db2 SVC dump processing is complete, z/OS issues message IEA911E to indicate whether enough space was available in the dump data set to contain the requested storage areas. If this message indicates that a partial dump was taken, but the 3.25 MB of summary storage is available in the dump, this dump is probably enough for problem diagnosis. Otherwise, IBM Software Support might request that you re-create the problem if storage areas that are required for problem determination are not included in the dump.

**System databases storage requirements**

The Db2 installation process automatically creates system databases for the resource limit facility, DDL registration, and EXPLAIN tables that are created under the SYSIBM schema. The storage requirements for these databases depend on the amount of user data that they each contain.

**Archive log data sets storage requirements**

If you decide to store the archive log data sets on disk, you need to reserve enough space on the device. The active log data set and the BSDS are both written to the same location. Therefore, you must reserve enough storage for the active log and the BSDS.

**Recommendation:** You should store archive logs on disk, especially if Db2 will be running in data sharing mode.

The amount of space for the archive primary and secondary archive log space are calculated as follows:

The primary space (PRIQTY subsystem parameter) is set to the size of the active log data set. The secondary space (SECQTY subsystem parameter) is set to a quarter the size of the active log data set.

**Related concepts:**

- [Active log data sets storage requirements](#)
- [Bootstrap data sets storage requirements](#)
Related reference:

DSNTPC1: CLIST calculations panel 2

Profile tables storage requirements
You need to reserve disk space for profile tables.

The amount of required disk space is approximately 100 to 200 MB.

Related reference:

Profile tables (Db2 Performance)

Virtual storage requirements for address spaces
Db2 uses several types of private address spaces, and each type requires storage.
- Db2 distributed data facility (DFD) address space (DSN1DIST)
- IRLM address space (IRLMPROC)
- Db2 system services address space (DSN1MSTR)
- Db2 database services address space (DSN1DBM1)
- Db2 allied agent address spaces
- Db2 stored procedures address spaces (WLM-named)
- Db2 administrative task scheduler address space

Db2 also uses extended common service area (ECSA), 64-bit common storage, and the z/OS Shared Memory Facility.

You might notice that the sample jobs sometimes use a region size of 0 KB. This region size is meant to simplify the installation process in those particular cases. The following topics provide some recommendations about Db2 region sizes. These recommendations are based on average use under normal circumstances on typical systems. Your requirements might be quite different.

Db2 distributed data facility address space (DSN1DIST) storage requirements
The distributed data facility address space supports network communications with other remote systems and execution of database access requests on behalf of remote users.

Recommendation: Use the default region size of 0 KB. This address space is started as part of DDF initialization. The start-up procedure is DSN1DIST.

IRLM address space (IRLMPROC) storage requirements
Db2 uses the IRLM to manage locks. When row locking is used, the number of locks that Db2 acquires might increase, which might in turn increase the amount of storage that IRLM requires. The number of locks that are acquired is dependent on your application.

You can estimate the IRLM control block structure at 540 bytes per lock. IRLM no longer supports placing locks in ECSA. All IRLM locks are now placed in the IRLM private address space.

The PC and MAXCSA parameters are no longer used, but you must maintain them for compatibility reasons. You must specify the parameters and values, but their values are not used. The MAXCSA value must be in the range 0-9999. The amount of available storage for IRLM private control blocks, including locks, is determined by the operating system and site-specific IPL parameters. IRLM reserves approximately 10% of the available private storage to be used for must-complete lock requests.
Use the MODIFY irlmproc,STATUS,STOR command to view and monitor the amount of private storage that IRLM has available.

You can dynamically adjust the amount of below-the-bar private storage by using the MODIFY irlmproc SET,PVT command. This command changes only the monitoring threshold of private storage for IRLM.

You can dynamically adjust the limit for above-the-bar private storage by using the MODIFY irlmproc SET,MLT command. This command updates only the MEMLIMIT that z/OS uses to control the amount of above-the-bar storage that can be requested by an address space. A new value remains in effect until the next time IRLM is stopped and restarted or until the MODIFY command is issued successfully again.

Neither of these MODIFY commands change the physical amount of storage that the operating system assigns to the address space.

Enabling data sharing further increases the storage that IRLM requires.

Related reference:

START irlmproc (z/OS IRLM) (Db2 Commands)

Db2 system services address space (DSN1MSTR) storage requirements

A variety of system-related functions are performed in the Db2 system services address space. The system services address space needs less space than the database services address space.

Recommendation: Specify 0 KB for the system services address space. The default start up procedure is DSN1MSTR.

Db2 database services address space (DSN1DBM1) storage requirements

The Db2 database services address space is the largest Db2 address space.

This address space uses storage above the 2 GB bar. The default start up procedure is DSN1DBM1. First, plan for a minimum of 90 MB in this address space, with 2 MB below the 16 MB line.

Most modules reside in the 31-bit extended private area. Storage usage below the 16-MB line is typically less than 2 MB.

Allied agent address space storage requirements

In Db2, any user address space that attaches to Db2 is referred to as an allied agent address space. This address space can include Resource Recovery Services attachment facility (RRSAF), TSO attachment facility, IMS attachment facility, CICS attachment facility, and batch address spaces.

The size of the Db2 attachment facility code in the allied agent address space depends on which attachment facilities you use. TSO requires about 372 KB for the DSN command. CAF, IMS, and RRSAF each require about 36 KB for the Db2 attachment facility code. For all attachment facilities, except CICS Transaction Server and RRSAF, the Db2 attachment facility code must run below the 16 MB line of virtual storage. Applications can run above the 16 MB line.

Related concepts:

CICS Transaction Server for z/OS Db2 Guide
Administrative task scheduler address space storage requirements

Each Db2 subsystem has a coordinated administrative task scheduler address space that it can start by using a z/OS started task procedure.

Therefore, if there are many Db2 subsystems running on a single z/OS system, there is a separate administrative task scheduler with a separate name for each one.

Two instances of the same administrative task scheduler cannot run simultaneously. To avoid starting up a duplicate administrative task scheduler, at startup the administrative task scheduler checks all of the address spaces control block first to ensure that there is no address space other than itself with the same name. If another address space with the same name is already up and running, the administrative task scheduler that is starting up immediately shuts down with a console error message. The administrative task scheduler can check only the address spaces that are in the same system, not in the entire Sysplex.

The administrative task scheduler address space stays up, even when Db2 comes down.

WLM-established stored procedures address spaces storage requirements

WLM-established stored procedures address spaces are WLM-established address spaces that can provide multiple isolated environments for stored procedures.

Each WLM-established stored procedures address space is associated with a Workload Manager environment.

Db2 for z/OS stored procedures support both main programs and subprograms; this support requires additional storage for each task control block (TCB). However, because you can run fewer programs in an address space, you can use less storage below the 16 MB line in each address space.

The amount of space that is required by a WLM-established stored procedures address space depends on the applications. Stored procedures that internally store a large amount of in-process work require more space than those that do not store much in-process work. Also, Java stored procedures require more space than others, such as COBOL stored procedures.

The overall amount of required space depends on the number of WLM-established stored procedures address spaces that are running, and the amount of space that is required for each.

Related tasks:

- Setting up a WLM application environment for stored procedures during installation
- Setting up a WLM application environment for stored procedures during migration
- Refreshing WLM application environments for stored procedures (Db2 Administration Guide)
- Maximizing the number of procedures or functions that run in an address space (Db2 Performance)
Shared memory storage requirements

*Shared memory* is a type of virtual storage that allows multiple address spaces to easily address storage that is common to Db2 address spaces. This memory resides above the 2 GB bar. Db2 also uses 64-bit common storage, which is shared by the entire LPAR, for control structures that need to be accessed while running processes in and out of the ssnmMSTR, ssnmDBM1, and ssnmDIST address spaces.

The shared memory object is created at Db2 startup, and all Db2 address spaces for the subsystem (DSN1DIST, DSN1DBM1, DSN1MSTR, and Utilities) are registered with z/OS to be able to access the shared memory object.

To define the size of the shared memory and the 64-bit common area, use the HVSHARE and HVCOMMON parameters of the IEASYSxx member in the parmlib concatenation.

**Restriction:** Ensure that you have defined a high enough value for HVSHARE to satisfy all component requests for shared memory within the z/OS image. The default value is 510 TB. Db2 12 requires 1 TB of contiguous 64-bit shared private storage for each subsystem. This storage is virtual, and it is controlled by the z/OS HVSHARE parameter in IEASYSxx. This storage is not backed at allocation, only as it is used.

**PSPI**

IFCID 0225 has statistics that you can use to determine the amount of currently defined storage and how much of it is currently allocated.

**Related information:**

IEASYSxx (symbol definitions and IEASYSxx members) (MVS Initialization and Tuning Reference)

Common service area storage requirements

Some of the Db2 load modules and control blocks are in common storage. Most of the space is in the extended common service area (ECSA) and 64-bit common storage.

With few exceptions, the CSA-resident load modules are link-edited with the residency attribute of RMODE(ANY). Most of the modules reside in ECSA (above the 16 MB line of virtual storage), as do most of the global control blocks. The IRLM lock control blocks (RHB and RLB) use 64-bit private storage.

Monitor your use of CSA and ECSA, and increase those values if necessary. By monitoring CSA below the 16-MB line, you can determine whether you need to increase the size of the ECSA.

When you IPL z/OS, you can override the CSA size with this syntax:

```
CSA=(a,b)
```

where:

- *a* is the number of kilobytes of CSA storage below the 16 MB line
- *b* is the number of kilobytes of ECSA storage above the 16 MB line

The CSA value is rounded down to the next 1 MB boundary, while the ECSA value is rounded up to the next 1 MB boundary.
To define the size of the 64-bit common area, you can use the HVCOMMON parameter of the IEASYSxx member in the parmlib concatenation.

Db2 12 requires 64-bit common storage. Db2 requires that at least 6 GB of 64-bit common storage is defined for each Db2 subsystem that runs on a z/OS image. Additionally, the 64-bit common storage must be large enough to accommodate the size of the log buffers that is specified by the OUTBUFF subsystem parameter plus 15%.

Restriction: Ensure that you have defined a high enough value for HVCOMMON to satisfy requirements for all Db2 subsystems plus all other use on the LPAR. The default value is 66 GB. The 6 GB of 64-bit common storage that is requested for each member must be contiguous.

Related reference:
- Common Service Area (CSA/Extended CSA) (MVS Initialization and Tuning Guide)

Calculating the storage requirement for the common service area:

You need to calculate the approximate storage requirement for the common service area (CSA). Under normal conditions, your virtual storage requirements in the CSA (below the 16-MB line) are unlikely to increase.

About this task

In a data sharing environment, if an IRLM performs member recovery or structure rebuilding, additional virtual storage is required. Any other increase in the amount of virtual storage that is needed occurs within the extended private area of the Db2 database address space and the extended private area of the distributed data address space.

Procedure

To calculate the approximate residual requirement for CSA (below the 16-MB line):
1. Start with up to 40 KB for each Db2 subsystem.
2. Add 24 KB for each started IRLM.

Calculating the storage requirement for the extended common service area:

You need to calculate the approximate requirement for the extended common service area (ECSA).

Procedure

To estimate storage that is needed for ECSA (above the 16-MB line) for each Db2 subsystem:
1. Use the following calculations for the local Db2 subsystem.
   a. Start with 5 MB of ECSA for the base.
   b. Add 4 KB for each local user. You can use four times the value of MAX [USERS field (THREAD subsystem parameter)]
   c. Add 5 MB for each IRLM subsystem. Each IRLM subsystem uses 4 - 5 MB of ECSA when no workload is active and the default trace settings are used. As the workload increases, the ECSA use in IRLM for locking might also increase, depending on the type of requests processed by IRLM.
For example, if a workload processes a large number of queries or notifies through IFI calls, the ECSA increases depending on the number of locks held, number of threads, number of members in the data sharing group, and so forth. When multiple IFI requests run concurrently, the ability of IRLM to reuse and release the ECSA storage is reduced. In such scenarios, the ECSA usage by IRLM might go up to 10 MB.

d. Add memory for IRLM optional trace buffers:
   - If optional tracing is used with the TRACE=YES parameter, add 1.3 MB.
   - If optional tracing is used with TRACE=\(n\), where \(n\) is the number of optional trace buffers, add \((512 \times n)\) KB.

e. Add 4 MB for instrumentation facility interface (IFI) buffers as requested by the monitoring programs. This includes ECSA storage for standard trace classes for accounting (1, 2, 3, 7, and 8) and statistics (1, 3, 4, 5 and 6). The memory requirement might go up to a maximum of 25 MB, if additional traces that are turned on, such as IFCID 003 or IFCID 148 with more than 10 active buffer pools or more than 10 active group buffer pools, IFCID 124 (SQL statement text), IFCID 199 (Buffer pool stats) and so forth.

2. If you use the distributed data functions of Db2 (DDF), you might find that you need more virtual storage. You can estimate how much your storage needs are likely to increase in the ECSA above the 16-MB line by adding the following amounts:

a. Add 1 KB for each SNA network connection. In z/OS 2.2 or higher, TCP/IP connections use zero ECSA storage. However, if you are uncertain of how many concurrent SNA connections you might have, or if you operate with z/OS 2.1 or earlier, use the value for \[[MAX REMOTE CONNECTED field (CONDBAT subsystem parameter)]\].

b. Add 2 KB for each thread that uses distributed processing. You can use two times the value for \[[MAX REMOTE ACTIVE field (MAXDBAT subsystem parameter)]\].

c. Add 1 KB for each Db2 site in your network, regardless of the number of connections between the site and the Db2 subsystem. Site refers to the following types of adjacent partners:
   - Downstream servers that are listed in communication database of your Db2 subsystem (as a TCP/IP address or an SNA LU name)
   - Upstream clients for which this Db2 subsystem acts as the DRDA server.

d. Add 2.5 MB for the code that relates to distributed processing.

**What to do next**

Specify this sum or a value that is larger than this sum as the second value of the CSA parameter of the IEASYS\(xx\) z/OS logical PARMLIB member. The logical PARMLIB is usually referred to as SYS1.PARMLIB. Specifying values that are too high is preferable to specifying values that are too low; making your values too low can result in a need to IPL z/OS. For example, if the ECSA size is too small, z/OS places the Db2 global load modules and control blocks for Db2 in CSA below the 16-MB line instead of above it. This can cause problems with coexisting z/OS subsystems.

Distributed threads are represented by z/OS SSRB control blocks which reside in the extended system queue area (ESQA) when they are paused. The amount of storage that is used might vary depending on the z/OS release that is used. Estimate 4 KB of ESQA for every active distributed thread. You can use \((4 \times \text{MAXDBAT})\) KB.
Virtual storage requirements for storage pools and working storage

During the ISPF tailoring session, you specify values that the DSNTINST CLIST uses to calculate main storage size.

**Recommendation:** Determine these values based on your estimated application workload before you install or migrate Db2.

These values provide an estimate of the private area that is needed by the DSN1DBM1 address space, the largest of the Db2 address spaces. If the estimated virtual storage for the address space is not available, you can re-evaluate the sizes that you requested.

The calculations in this topic are planning estimates. The noted values do not provide the exact limits, but they indicate a reasonable range of values.

The sum of the following values must fit the region size that Db2 supports:
- VSAM data set control block storage size
- Working storage size

The CLIST adds a fixed code size to the sum of these values to determine the main storage size.

After you specify the values listed above, the CLIST determines the EDM pool size and the size that is needed for the data set control blocks. The CLIST adds the working storage size and the fixed code size to update the region size that is used in the Db2 startup procedures. The CLIST also displays this information on CLIST calculations panel 1: DSNTIPC.

Use the formulas in this topic to estimate your storage needs. For your reference, the default values are included where appropriate.

**Related concepts:**
- Db2 database services address space (DSN1DBM1) storage requirements

**Related tasks:**
- Calculating virtual storage requirements for a dictionary (Db2 Performance)

**Related reference:**
- DSNTIPC: CLIST calculations panel 1

Calculating buffer pool size

*Buffer pools* are areas of virtual storage that are used to satisfy the buffering requirements for one or more table spaces or indexes. All Db2 subsystems use virtual buffer pools, which are backed by central storage or auxiliary storage. Buffer pools are created above the 2 GB bar.

**About this task**

*Virtual buffer pools:* For best results, use at least 100 KB of buffer pool space for each concurrent user. A value of 300 KB or more is recommended for improved performance. Very simple SQL statements that access small amounts of data can require less than this amount. Complex SQL statements that access large amounts of data can require more than this amount. Dynamic SQL statements might require a dynamic bind, which requires additional buffers for catalog tables and indexes. If an SQL statement requires directory access, additional buffers are also required for
directory page sets. UPDATE, DELETE, and INSERT statements require more buffers to keep updated pages in a buffer pool.

To efficiently support prefetch, allow 400 KB for each object that is concurrently accessed. For example, if you have a SELECT statement with a single table space scan, allow 400 KB for that statement. For another example, suppose you have a SELECT statement with a three-table sort merge join that uses an index for each table, and a parallel degree of 4. To obtain a rough estimate, calculate the total number of objects that are accessed and multiply the number of objects by 400 KB and then by the degree of parallelism. In this example, the equation would be (3 tables + 3 indexes + 1 work file for the sort) x 400 KB x 4. The result is 11200 KB that is recommended for this statement. Typically, these objects should be defined in separate buffer pools to provide better performance control. For example, you should have at least three separate buffer pools: one for indexes, one for tables, and one for work files.

During installation, you can set the buffer pool sizes on the installation panels. Later, you can use the ALTER BUFFERPOOL command to alter the sizes and other attributes of as many as 50 buffer pools for 4-KB page sets, 10 buffer pools for 8-KB page sets, 10 buffer pools for 16-KB page sets, and 10 buffer pools for 32-KB table spaces. The ALTER BUFFERPOOL command can make the changes dynamically while Db2 is running.

Important: Do not allocate more storage for buffer pools than available real storage for buffer pools. If you attempt to use more than the available real storage, performance will suffer.

Db2 limits the total amount of storage that is allocated for virtual buffer pools to approximately twice the amount of real storage. If you specify more than this amount for virtual buffer pools, Db2 allocates buffer pools during startup until twice the amount of real storage is used. Db2 then allocates the remaining buffer pools as follows:

<table>
<thead>
<tr>
<th>Page size</th>
<th>Number of pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 KB</td>
<td>2000</td>
</tr>
<tr>
<td>8 KB</td>
<td>1000</td>
</tr>
<tr>
<td>16 KB</td>
<td>500</td>
</tr>
<tr>
<td>32 KB</td>
<td>250</td>
</tr>
</tbody>
</table>

After these storage limits have been reached, you cannot increase the amount of virtual buffer pool storage unless you increase the amount of real storage that is available to the z/OS image.

Procedure

To calculate the virtual buffer pool sizes for your subsystem:

Use the following table.

<table>
<thead>
<tr>
<th>Virtual buffer pool calculation</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffers for BP0</td>
<td>20,000 x 4 KB = 80,000 KB</td>
</tr>
<tr>
<td>Buffers for BP1</td>
<td>+ 0 x 4 KB = 0 KB</td>
</tr>
<tr>
<td>Buffers for BP2</td>
<td>+ 0 x 4 KB = 0 KB</td>
</tr>
</tbody>
</table>

Table 1. Virtual buffer pool size calculation
Table 1. Virtual buffer pool size calculation (continued)

<table>
<thead>
<tr>
<th>Virtual buffer pool calculation</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffers for BP49</td>
<td>+____ x 4 KB = _____</td>
</tr>
<tr>
<td>Buffers for BP8K0</td>
<td>+____ x 8 KB = _____</td>
</tr>
<tr>
<td>Buffers for BP8K1</td>
<td>+____ x 8 KB = _____</td>
</tr>
<tr>
<td>Buffers for BP16K0</td>
<td>+____ x 16 KB = _____</td>
</tr>
<tr>
<td>Buffers for BP16K1</td>
<td>+____ x 16 KB = _____</td>
</tr>
<tr>
<td>Buffers for BP32K</td>
<td>+____ x 32 KB = _____</td>
</tr>
<tr>
<td>Buffers for BP32K9</td>
<td>+____ x 32 KB = _____</td>
</tr>
<tr>
<td></td>
<td>= _____</td>
</tr>
</tbody>
</table>

Related concepts:
- Introduction to Db2 data sharing (Db2 Data Sharing Planning and Administration)
- Changing the size of the group buffer pool (Db2 Data Sharing Planning and Administration)

Related tasks:
- Deciding how many buffer pools to use (Db2 Performance)
- Choosing buffer pool sizes (Db2 Performance)

Related reference:
- DSNTIP1: Buffer pool sizes panel 1
- DSNTIP2: Buffer pool sizes panel 2

Calculating sort pool size

The Db2 sort process uses two kinds of storage: local storage and buffer pool storage. Sort pool storage and buffer pools have storage above the 2 GB bar.

Related reference:
- MAX DATA CACHING field (MXDTCACH subsystem parameter)

Calculating sort pool storage in local storage:

The sort process creates fixed-length storage pools in local storage for internal sort structures and work areas. Local storage is created above the 2 GB bar at allocation time.

About this task

The Db2 sort work area (in-memory) has the following storage boundaries for each concurrent sort operation:
- Minimum sort storage = 240 KB
- Maximum sort storage = 128,000 KB
Db2 initially allocates storage for each sort and gradually adds more storage until the maximum sort work area limit is reached or the maximum number of nodes (32,000) are populated at the bottom of the sort tree, whichever occurs first. With 32,000 nodes at the bottom of the sort tree, the average run size for each sorted string is 64,000 records.

The default size of the sort pool is 10,000 KB, but you can override this default value by entering a sort pool size on installation panel DSNTIPC.

Procedure

To estimate the required storage for a sort pool:

Use the following formula:

\[ 32000 \times (16 + \text{sort key length} + \text{sort data length}) \]

Related tasks:

- Improving the performance of sort processing (Db2 Performance)

Related reference:

- DSNTIPC: CLIST calculations panel 1

Adjusting buffer pool storage that is used for sort work files:

Sort processing uses pages in the Db2 buffer pool for its initial input, for work files that contain intermediate results, and for the final output.

About this task

The buffer pools are not always dedicated to sort work files. The amount of sort activity determines how much the buffer pools are used. For heavy sort activity, sort records that have been written to the work files are temporarily written to the disk until buffer pool space becomes available.

Db2 considers the buffers that it uses for work files as sequentially accessed pages.

Procedure

To adjust the percentage of the buffer pool that is used for work files:

Use the VPSEQT parameter of the ALTER BUFFERPOOL command. If a buffer pool is used only for sort work files, you might set VPSEQT to 99%. If the buffer pool is used for sparse indexes or declared global temporary tables, set VPSEQT to 90%. If you do not have enough allocated storage to complete sort processing, you must allocate more disk space for the work file database.

Related concepts:

- Work file database storage requirements

Calculating RID pool size

The RID pool is an area of local storage that is reserved for record identifier (RID) sort processing, including RID list sort processing.
About this task

The RID pool is created at startup, but no space is allocated until RID storage is needed. When RID storage is needed, it is allocated above the 2 GB bar in 32 KB blocks, which are known as RID blocks.

Consider the following when you calculate RID pool size:

- Regardless of the current RID pool usage, each agent that requests a new RID map and RID list is granted two (32 KB) RID blocks (1 RID map and 1 RID list).
- If a thread needs another RID list block to be added to the existing RID map and RID list, the next 32 KB block is granted only if the RID pool usage is below the maximum RID pool size. It is possible for the RID pool size to be larger than the setting of the MAXRBLK subsystem parameter because every new agent is granted at least one RID map and one RID list.
- Startup RID storage = 0 (but acquired in 32 KB blocks as needed)

The default size of the RID pool is 400,000 KB, but you can override this default value by entering a RID pool size on installation panel DSNTIPC.

Procedure

To estimate the required storage for the RID pool:

Use the following formula:

\[
\text{number of concurrent RID processing activities} \times \text{average number of RIDs} \times 2 \times 8 \text{ (bytes per RID)}
\]

Related tasks:
- Managing RID pool size (Db2 Performance)

Related reference:
- RID POOL SIZE field (MAXRBLK subsystem parameter)

Calculating EDM pool sizes

The environmental descriptor manager (EDM) controls areas of virtual storage that hold cached objects. These areas are called pools, and thus the term EDM pool is frequently used to describe these areas of virtual storage.

About this task

In Db2 12, the EDM pools are the prepared-statement cache pool, the skeleton pool for packages, and the database descriptor (DBD) pool. All of these pools are located above the 2-GB bar.

When you cache prepared statements, they are stored in the prepared-statement cache along with static SQL statements. The number of prepared statements that are stored in the cache depends on the characteristics of the dynamic SQL statements that your application executes. One type of application typically benefits from caching prepared statements, whereas the other type usually does not. The first type of application uses dynamic SQL statements that are embedded in an application and used repeatedly. Applications and queries with this type of SQL statements benefit most from caching prepared statements because the statement can be used from the cache. However, applications that contain SQL statements that are infrequently used pay the cost of being added to the cache. For example, queries from QMF™ are likely to be prepared and executed only once. Caching prepared statements does not benefit applications that extensively use this kind of SQL statement.
The size of the prepared-statement cache pool is affected by the number of concurrent threads, the maximum number of DBATs, the number of plans, and the amount of estimated space for plans. If you use dynamic SQL, you need more working storage and less EDM pool space than if you use static SQL.

The skeleton pool for packages is used to store skeleton copies of packages (SKPT). The size of this pool is determined by the maximum number of unique plans in the EDM pool, the average statement size, and the average number of executed statements.

The EDM pool space for database descriptors is determined based on an estimated number of concurrently open databases and the average size of the database descriptor.

For Db2 12 installation, the CLIST provides initial EDM pool size settings on the CLIST calculations panel 1: DSNTIPC. These calculations are based on your site size. You might need to adjust these settings according to the specific needs of your site.

For migration to Db2 12, the CLIST reuses the values from your Db2 11 settings. However, you might need to adjust these values in Db2 12 if your workloads have grown.

**Procedure**

- If you are migrating to Db2 12, start with the EDM pool sizes that you used in Db2 11, and adjust the sizes as necessary.

- If you are installing Db2 12, the installation CLIST provides initial EDM pool size settings on the CLIST calculations panel 1: DSNTIPC. The calculations are based on your site size. You might need to adjust the settings according to the specific needs of your site.

**Related reference:**
[DSNTIPC: CLIST calculations panel 1](#)

**Calculating data set control block storage**

Calculate the data set control block storage requirements by determining the total number of open data sets (DSMAX).

**Procedure**

To determine the main storage requirements for your data set control blocks:

- Let $concdb$ be the number of concurrent databases, which is specified on installation panel DSNTIPE.

- Let $tables$ be the number of tables in each database, which is specified on installation panel DSNTIPD.

- Let $indexes$ be the number of indexes in each table. The installation CLIST sets this variable to 2.

- Let $pctpts$ be the percentage of partitioned table spaces.

- Let $avgpart$ be the average number of partitions in each partitioned table space.

- Let $partts$ be the average number of partitioned table spaces per database.

- Let $nonparti$ be the number of non-partitioning indexes per partitioned table space.

1. Calculate the total number of open tables with the following formula:

   \[ \text{opntab} = concdb \times tables \]
2. Calculate the number of open data sets for partitioned table spaces with the following formula:

\[
\text{opnptsds} = \text{opntab} \times \left(\frac{\text{pctpts}}{100}\right) \times \text{avgpartconcdb} \times \left(\frac{\text{tables} \times \text{indexes} + \text{tblspaces}}{\text{avgpart} \times 2 \times \text{partts}} - \frac{2 \times \text{partts}}{\text{partts}}\right)
\]

3. Calculate the number of open data sets with the following formula:

\[
\text{concdb} \times \left(\frac{\text{tables} \times \text{indexes} + \text{tblspaces}}{\text{avgpart} \times 2 \times \text{partts}} - \frac{2 \times \text{partts}}{\text{partts}}\right)
\]

You can modify DSMAX by editing job DSNTIJUZ. The maximum number of concurrently open data sets that is allowed by Db2 is 200,000.

4. To calculate the main storage that is required for your data set control blocks, use the following formula:

\[\text{DSMAX} \times 5 \text{ KB}\]

**Example**

The default, as calculated by the DSNTINST CLIST, is

\[20000 \times 5 \text{ KB} = 100000 \text{ KB}\]

This method of calculation ignores partitioned table spaces and partitioning index spaces. It also assumes that all data sets in the database are open if the database is in use. You could enter a smaller value for the number of concurrent databases if only a few of the data sets in a database are typically opened. The larger the value of DSMAX, the longer data sets stay open.

**Recommendations:**

- Move the scheduler work area (SWA) above the 16 MB line of z/OS virtual storage by using JES initialization statements, JES exit routines, or the SMF exit routine (IEFUJV). This way, you can save approximately 1 KB for each open data set in virtual storage below the 16 MB line and avoid potential storage errors. To determine the amount of storage that is needed below the 16-MB line, use 0.016 KB for the multiplication factor if the SWA is above the line or 1.2 KB for the multiplication factor if the SWA is below the line. The preceding calculations and the calculations in the CLIST presume that the SWA is above the line.

- If you do not move the SWA above the 16 MB line, you should not use a DSMAX value greater than 5000.

**Related concepts:**

- How Db2 determines the initial value of DSMAX (Db2 Performance)

**Calculating working storage requirements**

Working storage is private virtual storage in the Db2 database services address space, which is primarily above the 2 GB bar. Working storage is used to hold temporary data that is needed for the processing of an SQL request.

**About this task**

If you use static SQL, you have larger plan sizes, typically varying from 15 KB to 1 MB. Typical sites would use about 300 KB for each thread of working storage for dynamic SQL users, and 100 KB per thread for static SQL users. A thread is a structure that describes an application connection to Db2. The CLIST does not include information about open compressed table spaces. Compression dictionaries are stored above the 2-GB bar. Therefore, if you use compressed table spaces, you need additional storage.
Procedure

To estimate the amount of working storage:

1. Start with 600 KB.
2. Determine the amount of space in KB that is needed for active users.
   - Start with 11,920 bytes for each concurrent Db2 user. This value is specified by the CTHREAD subsystem parameter.
   - Add 11,920 bytes for each remote Db2 user. This value is specified by the MAXDBAT subsystem parameter.
   - Divide the total by 1,024 to convert to KB.
3. Determine the amount of space in KB that is needed for inbound DDF connections.
   - Start with 298 bytes per connection.
   - Divide the number of bytes by 1,024 to convert to KB.
   - Subtract the result from the maximum number of concurrent inbound DDF connections. This value is specified by the CONDBAT subsystem parameter.
   - Multiply the result by 4.
4. Estimate the working storage by adding the results from steps 1, 2, and 3

$$600 + (((CTHREAD + MAXDBAT) \times 11920)/1024) + \left((MAXDBAT - (200 \times 298)/1024) \times 4\right)$$

Example

The default, as calculated by the DSNTINST CLIST, is:

$$600 + (((200 + 200) \times 11920)/1024) + \left((10000 - (200 \times 298)/1024) \times 4\right) = 45024$$

Related concepts:
- How Db2 allocates allied threads (Db2 Performance)

Related tasks:
- Compressing your data (Db2 Performance)

Related reference:
- MAX USERS field (CTHREAD subsystem parameter)
- MAX REMOTE CONNECTED field (CONDBAT subsystem parameter)
- MAX REMOTE ACTIVE field (MAXDBAT subsystem parameter)

Calculating virtual storage below the 16-MB line

This calculation produces an estimate of virtual storage constraints below the 16-MB line in the Db2 database services address space.

About this task

Most of the needed virtual storage is in extended private storage and storage above the 2 GB bar, including the buffer pool, the EDM pool, and almost all of the code and working storage. This is the difference between the total storage and the estimated region size. The estimated region size does not include extended private storage or storage above the 2 GB bar. It includes only the data set control block storage size and some of the code.
Procedure

To estimate the size of storage below the 16-MB line:

Use the following formula:

\[ 600 \text{ KB} + \left( \frac{(\text{MAX USERS} + \text{MAX REMOTE ACTIVE}) \times 298}{1024} \right) + (\text{DSMAX} \times 0.016) \]

Example

The default, as calculated by the DSNTINST CLIST, is

\[ 600 \text{ KB} + \left( \frac{(200 + 200) \times 298}{1024} \right) + (20000 \times 0.016 \text{ KB}) = 1036 \text{ KB} \]

If the scheduler work area (SWA) is above the 16-MB line, multiply the number of data sets by 0.016 KB; if the SWA is below the 16-MB line, multiply the number of data sets by 1.2 KB.

The CLIST calculations panel, DSNTIPC, displays storage sizes that are calculated by the DSNTINST CLIST.

Related reference:

DSNTIPC: CLIST calculations panel 1

Calculating real storage requirements

Db2 can use real storage to reduce I/O and processor times and to improve response time and throughput. The amount of real storage that Db2 needs varies greatly.

About this task

Some users find that their organizations need several times the estimates listed below, whereas others need less. The amount of storage is an important parameter in Db2 performance. Performance monitoring programs give you a more accurate estimate of your storage requirements than the formulas in this topic because these programs can take specific environmental characteristics into account.

For the Db2 buffer pools, the EDM pool, log buffers, and working storage, the amount of real storage must be the same as the amount of virtual storage. Paging activity in the buffers is an indication of a problem. If you do not have enough real storage, you might need to reduce your buffer pool size, which allows fewer concurrent users. You also need space to contain locks, the working set of code in all address spaces, log buffers, and common storage space. In Db2 12, log buffers are permanently page-fixed. Therefore, the entire size of the output buffer needs to be used to estimate real storage. Because some of the figures that are used in virtual storage calculations are maximums, whereas the real storage figures typically use activity for the peak, changes are needed in the calculations. The virtual storage figures concentrate on the most constrained address space, but real storage work must include them all. For more information about each category, see the information specified in the first two tables below.

Some storage for castout engines or DASD read and write of a compressed index might be fixed through a PAGEFIX operation. More castout engines or more compressed indexes can cause more storage to be permanently fixed. MVS Real Storage Manager (RSM) controls where the PAGEFIX occurs in real storage, and it can occur below the 16 MB line. If you have concerns about this behavior, consult with IBM Support for MVS RSM.
Procedure

To estimate real storage requirements:

1. Multiply the virtual size of each storage area by the factor that is indicated in the following table to estimate the real size for that area.

<table>
<thead>
<tr>
<th>Category</th>
<th>Default Virtual size</th>
<th>× Factor</th>
<th>= Real size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer pools</td>
<td>104,200 KB</td>
<td>1.0</td>
<td>104,200 KB</td>
</tr>
<tr>
<td>Sort pool</td>
<td>10,000 KB</td>
<td>0.5</td>
<td>5000 KB</td>
</tr>
<tr>
<td>RID pool</td>
<td>400,000</td>
<td>0.5</td>
<td>200,000</td>
</tr>
<tr>
<td>EDM pool</td>
<td>33,600 KB</td>
<td>1.0</td>
<td>33,600 KB</td>
</tr>
<tr>
<td>Data set size</td>
<td>17,928 KB</td>
<td>0.6</td>
<td>10,756.8 KB</td>
</tr>
<tr>
<td>Code size + 1100 (4 address spaces)</td>
<td>40,000 KB</td>
<td>0.5</td>
<td>20,000 KB</td>
</tr>
<tr>
<td>Working storage + (DSCF + DDF)</td>
<td>55,800 KB</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Log buffers</td>
<td>4000 KB</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Lock space</td>
<td>5000 KB</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>Common storage</td>
<td>2160 KB</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

Total real storage size =

2. Plan to add real storage for additional users.

The following table uses rough estimates to approximate the amount of additional real storage that is needed by several kinds of users. If you have more concurrent users, plan to add real storage.

<table>
<thead>
<tr>
<th>Type of user</th>
<th>Additional real storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction</td>
<td>150 KB</td>
</tr>
<tr>
<td>Query</td>
<td>400 KB</td>
</tr>
<tr>
<td>Batch</td>
<td>700 KB</td>
</tr>
</tbody>
</table>

**Recommendation:** If you are assessing the storage needs of a running subsystem, look at the Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS report for IFCID 0225. This report calculates the virtual storage per user.

3. Plan to add 28 KB of real storage for each open index partition.

Example

The following table shows the default real storage size calculations.

<table>
<thead>
<tr>
<th>Category</th>
<th>Default Virtual size</th>
<th>× Factor</th>
<th>= Real size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer pools</td>
<td>104,000 KB</td>
<td>1.0</td>
<td>104,000 KB</td>
</tr>
<tr>
<td>Sort pool</td>
<td>10,000 KB</td>
<td>0.5</td>
<td>5000 KB</td>
</tr>
<tr>
<td>RID pool</td>
<td>400,000</td>
<td>0.5</td>
<td>200,000</td>
</tr>
<tr>
<td>EDM pool</td>
<td>33,600 KB</td>
<td>1.0</td>
<td>33,600 KB</td>
</tr>
<tr>
<td>Data set size</td>
<td>17,928 KB</td>
<td>0.6</td>
<td>10,756.8 KB</td>
</tr>
</tbody>
</table>
### Table 4. Default real storage size calculation (continued)

<table>
<thead>
<tr>
<th>Category</th>
<th>Default virtual size</th>
<th>× Factor</th>
<th>= Real size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code size + 1100 (4 address spaces)</td>
<td>40,000 KB</td>
<td>0.5</td>
<td>20,000 KB</td>
</tr>
<tr>
<td>Working storage + (DSCF + DDF)</td>
<td>55,800 KB</td>
<td>1.0</td>
<td>55,800 KB</td>
</tr>
<tr>
<td>Log buffers</td>
<td>4000 KB</td>
<td>1.0</td>
<td>4000 KB</td>
</tr>
<tr>
<td>Lock space</td>
<td>5000 KB</td>
<td>0.4</td>
<td>2000 KB</td>
</tr>
<tr>
<td>Common storage</td>
<td>2160 KB</td>
<td>0.4</td>
<td>864 KB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-----------</td>
</tr>
<tr>
<td><strong>Total real storage size</strong></td>
<td></td>
<td></td>
<td><strong>436,021 KB</strong></td>
</tr>
</tbody>
</table>

### Related tasks:
- Calculating buffer pool size
- Calculating EDM pool sizes
- Calculating working storage requirements

### Planning for performance of Db2

When you install or migrate Db2, consider your performance goals for your subsystem or data sharing member. Use z/OS Workload Manager (WLM) to define these goals.

Also, consider the following guidance:
- Monitoring and managing performance are more difficult with few plans.
- Never-ending threads and plans reuse accumulate thread storage and working storage.
- Each distributed COMMIT statement needs package accounting.
- Consider reusing threads.
- Finding unused packages and cleaning up mistakes are difficult unless ownership is known.
- Performance monitoring may require analyzing everything in a CICS region instead of just one plan.
- Plan is a required short name.
- Package name is optional, long, multipart, and not always easy to find.
- Consider differentiating by plan within a CICS region and having at least 5 to 20 plans within a region. You can work faster and make changes on a more granular basis.
- Eligible processes can be offloaded to IBM Z Integrated Information Processor (zIIP) specialty engines to reduce the load on general purpose processors.

### Related concepts:
- [z/OS performance options for Db2 (Db2 Performance)]
- [IBM Z Integrated Information Processor (zIIP) specialty engines (Db2 Performance)]

### Related tasks:
- [Controlling resource usage (Db2 Performance)]
Planning for release coexistence in Db2 12

In release coexistence, applications can continue to access Db2 data while you migrate data sharing members or distributed subsystems. However, you must weigh the benefit of continuous availability against the operational costs of running in coexistence. Most new capabilities and enhancements in Db2 12 remain unavailable during coexistence, and additional system management considerations apply.

Db2 12 supports release coexistence with Db2 11 in data sharing and distributed environments, as long as function level 500 or higher has not been activated on any Db2 12 member or subsystem.

Important: Do not issue the ACTIVATE command or run job DSNTIJAF for activation of function level 500 or higher until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

For more information about release coexistence in Db2 data sharing, see Release coexistence in Db2 data sharing.

Related tasks:

Avoid automatic binds at migration to Db2 12
Chapter 3. Loading the Db2 libraries

IBM distributes Db2 on tapes or cartridges, depending on which feature you order. To load the Db2 libraries, use System Modification Program Extended (SMP/E). SMP/E processes the installation tapes or cartridges and creates Db2 distribution libraries, Db2 target libraries, and SMP/E control data sets.

For information on using SMP/E, see Db2 Program Directory.

Related tasks:
Tailoring Db2 12 installation and migration jobs with the CLIST

Editing the SMP/E jobs

Before running any of the SMP/E jobs, you must edit them. Some of the items that you might want to edit include link list options and naming conventions for the Db2 library data sets.

Creating JOB statements

The SMP/E jobs do not include JOB statements. Although JOB statements are often built automatically, creating your own JOB statements that are correct for your site is usually easier than editing the provided JOB statements.

Procedure

To create JOB statements:

1. If you are using ISPF to edit and submit the SMP/E jobs, edit a member that contains the JOB statement. Delete all text except the JOB statement. Then use the ISPF COPY command to copy the member into each job before submitting it.

2. If you are using TSO to submit the SMP/E jobs, edit a JOB statement and submit that JOB statement with each job. For example, data set JCL.CNTL(J) might contain the following code:

   ```
   //DB2INST
   JOB ACCT,NAME,
   //   MSGCLASS=A,MSGLEVEL=(1,1),
   //   TIME=(10),USER=SYSADM,PASSWORD=xxxxxxxx
   /*JOBPARM ....
   /*ROUTE PRINT ....
   ```

   When you are ready to submit a job, use a command like the following command:

   ```
   SUBMIT (JCL(J) JCL(DSNxxxxx))
   ```

   In this command, xxxxxx represents the last five characters of the SMP/E job name. This command submits the JOB statement along with the job.

Choosing link list options

Before you run the SMP/E jobs to load the Db2 libraries, you can choose your link list options.
Procedure

To choose your link list options:

Consider and choose your link list options based on the following information about the load module libraries:

*prefix.SDSNLINK*

Contains Db2 modules that are referred to as early (ERLY) code, which must reside in an APF-authorized library. You must place early code modules in the link list because they are loaded at subsystem initialization during IPL. In the link list, you can have only one level of the early code. That early code can be shared by multiple subsystems and releases of Db2. Therefore, you must choose the level of early code that is appropriate for all subsystems that use the link list. If your system is at the prerequisite maintenance level, your Db2 11 early code is upward compatible with Db2 12. The Db2 12 early code is downward compatible with Db2 11. If you plan to run Db2 10 subsystems on the same LPAR as Db2 12, choose Db2 11 ERLY code.

If you are migrating, maintenance to early code or an installation of new early code requires that you activate the early code, which can be done in a couple of ways:

- Activate the early code by doing an IPL of z/OS.
- Another option to activate maintenance to early code is to issue the `REFRESH Db2,EARLY` command. If you use this command, the subsystem that is having its early code refreshed must be stopped when the command is issued.

To initialize a new Db2 subsystem, you can issue the z/OS SETSSI command, which will execute the early code initialization routine and update the subsystem vector table (SSVT).

You should schedule a z/OS IPL before or during a migration to a new release of Db2. This IPL is necessary because migration job DSNTIJMV makes changes to SYS1.PARMLIB that are not recognized by z/OS until the next IPL. Alternatively, you can use the SETSSI, SETPROG APF, and SETPROG LNKST commands to define the subsystem to z/OS until the next IPL. Changes that DSNTIJMV makes to the SYS1.PARMLIB might affect the following libraries:

- New subsystem definitions in IEFSSNx
- New APF libraries in IEAAPFx
- New load module libraries in LNKLSTx

*prefix.SDSNLOAD*

- Is a main load module repository
- Can be shared by multiple subsystems at the same release level
- Allows only Db2 to modify code
- Holds default exit routines
- Is APF-authorized
- Must be a PDSE

*prefix.SDSNLOD2*

Contains a PDSE data set, which contains JDBC and SQLJ DLLs. Although Db2 does not require that SDSNLOD2 be APF-authorized, be aware that if this data set is in a STEPLIB data set concatenation of an address space that does need APF authorization, SDSNLOD2 must also be APF-authorized.
prefix.SDSNEXIT

- Holds the subsystem parameter module, DSNHDECP, and user-written exit routines
- Is modified by user
- Is APF-authorized

Link list recommendation for IRLM
Adding the IRLM load module DXRRL183 to the link list is highly recommended so that IBM Support can assist in diagnosing and resolving potential problems. Although IRLM can operate without this load module, if the DXRRL183 load module is not in the link list, users cannot run CT TRACE commands to externally start and stop IRLM component traces, which are critical in diagnosing problems. To add DXRRL183 to the link list if it is not already there, you need to copy the module into a link-listed library. After you apply maintenance to IRLM that affects DXRRL183, remember to copy the updated module to the link list.

Integrated Cryptographic Services Facility link list requirement
If you plan to use encryption with your Db2 subsystem, you must include in the link list the Integrated Cryptographic Services Facility library that contains the SCSFMOD0 load module. The SQL statement SET ENCRYPTION PASSWORD and the following built-in functions require this support:
- ENCRYPT
- DECRYPT_BIT
- DECRYPT_DB
- DECRYPT_CHAR
- GETHINT
- HASH_MD5
- HASH_SHA1
- HASH_SHA256

Supporting one Db2 subsystem
You can choose among several methods of maintaining a single Db2 subsystem. The following steps describe what is probably the easiest method for most sites:

1. Change the SMP/E procedure DSNALLOC to assign all load modules to prefix.SDSNLOAD. You can do this by changing the data set name for DDDEF (SDSNLINK) from prefix.SDSNLINK to prefix.SDSNLOAD.
2. Remove the allocation for prefix.SDSNLINK from the allocation job DSNALLOC.
3. Include prefix.SDSNLOAD (instead of prefix.SDSNLINK) in the LNKLSTxx member of SYS1.PARMLIB.

Supporting multiple Db2 subsystems
Supporting multiple subsystems can mean several things. You can have two or more Db2 subsystems at the same release and service level (for example, two Db2 12 subsystems). Create separate libraries for DSNHDECP and user-written exit routines of each Db2 subsystem.

You can also have two or more Db2 subsystems at the same release level, but at different service levels. For example, you can have a Db2 12 production subsystem and a Db2 12 test subsystem at different service levels.
levels. Alternately, you can have two Db2 subsystems at different release levels. For example, you can have a Db2 12 subsystem and a Db2 11 subsystem.

In either of these cases, you can use an INCLUDE statement in JCL or a started task to identify the necessary load modules.

The Db2 subsystem must use the appropriate release level of prefix:SDSNLOAD, but the application attachment code (for example, CICS, CAF, or TSO attachment facilities) can use code that is either one release level down or one release level up from that of prefix:SDSNLOAD. To use application attachment code that is either one level down or one level up from that of prefix:SDSNLOAD, place the attachment code in a different STEPLIB data set from the STEPLIB data set that Db2 executes.

Related tasks:
Preparing for Db2 12 migration in Db2 data sharing
Migration step 17: Refresh the z/OS definition of Db2

Related reference:
-REFRESH DB2,EARLY (Db2) (Db2 Commands)
z/OS SETSSI Command (MVS System Commands)
z/OS SETPROG Command (MVS System Commands)

Related information:
Accessing the correct Db2 program library

If you do not place prefix:SDSNLOAD in the LNKLSTxx member of SYS1.PARMLIB, you must provide JOBLIB or STEPLIB statements for it in certain types of programs and procedures.

About this task

The installation and migration jobs that are provided with Db2 12 already contain the necessary JOBLIB or STEPLIB statements. In addition, the startup procedures that Db2 provides include STEPLIB statements for their respective program libraries, prefix:SDSNLOAD, and prefix:SDSNEXIT.

Procedure

To access the correct program library:

Provide STEPLIB or JOBLIB statements for the following types of programs and procedures if you do not place prefix:SDSNLOAD in the LNKLSTxx member of SYS1.PARMLIB.

- **TSO or batch jobs** that access Db2 services require JOBLIB or STEPLIB statements for prefix:SDSNLOAD. These jobs include TSO logon procedures and batch jobs that access the DSN command and subcommands, the Db2 precompiler, and Db2 utilities.

- **IMS control, message, and batch processing jobs** also require JOBLIB or STEPLIB statements for prefix:SDSNLOAD. You must specify the Db2 load library in the startup procedure for each IMS region (IMS control, message processing program (MPP), batch message processing (BMP), and Fast Path region) that can communicate with Db2. You can do this in two ways:
- If all the data sets that the JOBLIB or STEPLIB statement refers to for an IMS region are APF-authorized, add the DD statement for prefix.SDSNLOAD to the JOBLIB or STEPLIB statement. If you are using the DYNAM option of COBOL, the IMS RESLIB DD statement must precede the reference to prefix.SDSNLOAD in the JOBLIB or STEPLIB statement.

- If any of the data sets that the JOBLIB or STEPLIB statement refers to for the IMS region are not APF-authorized, add the DFSESL DD statement for prefix.SDSNLOAD. All libraries that are specified on the DFSESL DD statement must be APF-authorized. The DFSESL DD statement is not required by the Db2 DL/I batch support. IMS requires that an IMS RESLIB DD statement also be referred to by the DFSESL DD statement, as in the following example:

  //DFSESL DD DSN=ims_reslib,DISP=SHR
  // DD DSN=prefix.SDSNLOAD,DISP=SHR

- CICS procedures, including the CICS initialization JCL, also need to include Db2 libraries.

Related tasks:

- CICS startup JCL requirements for connection to Db2

Performance implications of installation libraries

The performance of your system might be degraded by including modules in the libraries that are included in the link list and you might want to consider the strategies below. These general suggestions might not match the specific needs of your site.

Adding many modules to the libraries that are included in the link list can reduce system performance. However, adding only a few modules to the libraries requires additional STEPLIB or JOBLIB statements. Because these STEPLIB or JOBLIB statements must be searched before the link list is searched, this approach can also reduce system performance. The approach that produces the best performance for your site depends on the environment in which you use Db2. Regardless of which attachment facilities you use, the modules in prefix.SDSNLINK must always be in the link library list.

If you run multiple Db2 subsystems on the same LPAR, but the Db2 subsystems are at different release and service levels, you need to include prefix.SDSNLINK, but not prefix.SDSNLOAD, in the LNKLSTxx member of SYS1.PARMLIB. Place the needed STEPLIB or JOBLIB statements in the JCL procedures and batch jobs.

If you are using Db2 with IMS, you need to include prefix.SDSNLINK, not prefix.SDSNLOAD, in the LNKLSTxx member of SYS1.PARMLIB because both the IMS RESLIB and prefix.SDSNLOAD have the DSNHLI alias. Place the needed STEPLIB or JOBLIB statements in the IMS procedures.

If you are using Db2 with IMS and you want prefix.SDSNLOAD (in addition to prefix.SDSNLINK) in the LNKLSTxx member of SYS1.PARMLIB, ensure that the library concatenation for prefix.SDSNLOAD and the IMS RESLIB are correct for your site because both libraries have the DSNHLI alias.

If you are using Db2 with CICS, you should probably put prefix.SDSNLINK, not prefix.SDSNLOAD, in the LNKLSTxx member of SYS1.PARMLIB. Then, place the needed STEPLIB or JOBLIB statements in the CICS procedures.
If you use the TSO attachment facility, RRS attachment facility, and call attachment facility, you might need to handle placement of load modules somewhat differently, as follows:

- If you use the DSN command and its subcommands infrequently, place **only prefix.SDSNLINK in the LNKLSTxx member of SYS1.PARMLIB. Provide the necessary STEPLIB or JOBLIB statements in your TSO logon procedures or in your JCL if you are using batch.**

- If you use the DSN command and its subcommands frequently, you might also want to move the TSO attachment facility load modules to a library that is defined in the LNKLSTxx. The TSO attach modules are DSNECP00, DSNECP10, DSNESM00, and DSNELI.

- If you use the call attachment facility (CAF) frequently, move the CAF load modules (DSNACAB, DSNACAF, and DSNALI) to a library that is defined in the LNKLSTxx.

- If you use the RRS attachment facility (RRSAF) frequently, move the RRSAF load modules (DSNARRS and DSNRLI) to a library that is defined in the LNKLSTxx.

- If you use the CAF, RRSAF, or the DSN command and its subcommands frequently, you can move the eligible load modules to a library that is defined in the link pack area (LPA), in IEALPAxx member of SYS1.PARMLIB. The CAF and DSN load modules must reside below the 16 MB line of z/OS virtual storage.
  - The TSO load modules that you can place in the LPA are DSNECP00, DSNECP10, DSNESM00, and DSNELI. If you include these modules in the LPA, remember to include the appropriate aliases for DSNECP00 (DSN) and DSNELI (DSNHLI).
  - The CAF load modules that you can place in the LPA are DSNACAF and DSNALI. If you include these modules in the LPA, remember to include the appropriate alias for DSNALI (DSNHLI2). Do not include DSNACAB in the LPA because it is a data-area-only, non-executable load module.
  - The RRSAF load modules that you can place in the LPA are DSNARRS and DSNRLI. If you include these modules in the LPA, you must include the appropriate alias for DSNRLI (DSNHLIR).

**Attention:** If modules are moved or copied from one library to another, you must make changes to SMP/E control data to reflect the movement. If you do not make these changes, future service or changes to the modules will not be processed correctly.

**Naming conventions for Db2 library names**

You need to modify the Db2 library data set names in the SMP/E jobs.

These data sets are listed in Table 1 of [What you produce](#). Their names are composed of three parts:

- A user-defined prefix
- A fixed base name: for example, SDSNLOAD
- An optional user-defined suffix

The Db2 12 default prefix (*prefix*) is used in this information; the default suffix is null. You need to edit each of the Db2SMP/E jobs and follow the directions in the header notes of each job to specify the names of the SMP/E data sets. If you want to add a suffix, edit the SMP/E procedures and allocation jobs. The prefix cannot exceed 18 characters. The suffix cannot exceed 17 characters, minus the length of
the prefix. In addition, any data set names that exceed eight characters must be in groups of no more than eight characters, separated by periods. The qualified data set name cannot exceed 44 characters.

You can also change the base name of these libraries or load them into another data set. If you do this, however, you might need to do additional editing of the installation or migration jobs. The DSNTINST CLIST, which you use later to tailor the installation and migration jobs, uses the following default data set names:

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix.SDSNCLST</td>
<td>Prefix Data Set CLIST</td>
</tr>
<tr>
<td>prefix.SDSNEXIT</td>
<td>Prefix Data Set EXIT</td>
</tr>
<tr>
<td>prefix.SDSNLINK</td>
<td>Prefix Data Set LINK</td>
</tr>
<tr>
<td>prefix.SDSNLOAD</td>
<td>Prefix Data Set LOAD</td>
</tr>
<tr>
<td>prefix.SDSNDBRM</td>
<td>Prefix Data Set DBRM</td>
</tr>
<tr>
<td>prefix.SDXRRESL</td>
<td>Prefix Data Set XIX</td>
</tr>
<tr>
<td>prefix.SDSNEXIT.</td>
<td>Exit data set</td>
</tr>
<tr>
<td>prefix.DBRMLIB.DATA</td>
<td>DBRMLIB data set</td>
</tr>
<tr>
<td>prefix.RUNLIB.LOAD</td>
<td>RUNLIB LOAD dataset</td>
</tr>
<tr>
<td>prefix.SRCLIB.DATA</td>
<td>SRCLIB data set</td>
</tr>
<tr>
<td>prefix.SDSNIVPD</td>
<td>IVPD data set</td>
</tr>
<tr>
<td>prefix.SDSNC.H</td>
<td>NC.H data set</td>
</tr>
</tbody>
</table>

Recommendation: Use the supplied naming convention.

Document any changes you make to the library names in the SMP/E jobs. You must specify these library names again during the ISPF tailoring session.

## Specifying SMP/E data set options

You have several options regarding how you establish and use SMP/E data sets.

### About this task

You must decide whether you want Db2 and IMS to share SMP/E data sets. You must also decide whether you need an additional set of SMP/E data sets. An additional set of SMP/E data sets is required if you support more than one release of Db2.

Related information:

[SMP/E for z/OS: User's Guide](#)

### Sharing SMP/E data sets with IMS

Db2 and z/OS cannot share SMP/E data sets because some module names and macro names are common to both products. Under certain conditions, however, Db2 can share SMP/E data sets with IMS.

### About this task

If you do not share SMP/E data sets with IMS, skip this step.

The allocation job DSNALLOC defines a new set of SMP/E data sets that Db2 and IMS are to share.

### Procedure

To define sharing of SMP/E data sets between Db2 and IMS:

You must modify your allocation job for either of the following situations:

- **Situation 1**: You decide to have separate SMP/E data sets for Db2 and IMS. In certain situations, Db2 and IMS cannot share SMP/E data sets. You must have separate SMP/E data sets if you want to have two IRLMs.
Even if you are not required to have separate SMP/E data sets, you might want to keep them separate. If Db2 and IMS share the SMP/E data sets, you need to accept or reapply Db2 corrective service to these data sets to allow IMS SYSGENs.

To establish separate SMP/E data sets for Db2 and IMS, change the data set prefix that your allocation job uses to a value other than the prefix that you use for your current IMS SMP/E data sets. The allocation jobs use the prefix “IMS”. Changing this prefix prevents the allocation job from replacing your current SMP/E data sets and still allows it to create new SMP/E data sets.

- **Situation 2**: You decide to share SMP/E data sets between Db2 and IMS, but you want to use the SMP/E data sets that already exist for IMS. To do this, remove the data set allocation and initialization statements from your allocation job. When you run the job, no SMP/E data sets are created, and Db2 then shares the existing SMP/E data sets with IMS.

### SMP/E data sets for two releases

A single set of SMP/E zone structures can record only one release of Db2. Maintaining separate zone structures for both Db2 12 and your previous release is strongly recommended until you are sure that you will not need to fall back.

The SMP/E jobs that are provided with Db2 assume that you will allocate a new set of SMP/E data sets for the new release. When you run your allocation job (DSNALLOC), it creates a set of SMP/E data sets. If you choose to reuse your Db2 11 zone structure, you can run job DSNTIJUD to delete SMP/E data for Db2 11. However, after you run this job, you cannot fall back to your previous release.

You can create an additional set of SMP/E data sets either by copying them from a prior release of Db2 or by allocating a new set. Allocating a new set is faster because no data must be deleted.

**Recommendation**: Copy a prior set so that you can then perform service regression checking.

### Specifying a new SMP/E prefix for IRLM

The SMP/E prefix in the SMP/E jobs is the same for the new IRLM as for the old IRLM. Consequently, if you do not change the SMP/E prefix, the jobs overwrite your old IRLM.

**Procedure**

If you do not want the jobs to overwrite your old IRLM:

Edit the SMP/E jobs accordingly.

### What you produce

During SMP/E processing, Db2 is loaded into the distribution and target libraries. Db2 uses the distribution libraries to maintain Db2 and contain the master copy of all elements for your Db2 subsystem.

The target libraries contain the various Db2 components. Db2 target libraries are updated when you apply corrective service.

The following table describes all the Db2 distribution libraries. The distribution libraries contain the master copy of all elements for your Db2 system.
Table 5. Db2 distribution libraries

<table>
<thead>
<tr>
<th>Distribution libraries</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix.ADSNBASE</td>
<td>This library contains all jobs that are required to complete SMP/E installation.</td>
</tr>
<tr>
<td>prefix.ADSNENU or ADSNDKF</td>
<td>This library contains the Db2 English or Kanji task panels, respectively.</td>
</tr>
<tr>
<td>prefix.ADSNLOAD</td>
<td>This library contains an individual object module for every Db2 load module. It contains the IRLM load modules if you choose to install IRLM into the same distribution libraries as Db2. This library must be a PDSE data set.</td>
</tr>
<tr>
<td>prefix.ADSNLOD2</td>
<td>This library contains a PDSE data set, which contains JDBC and SQLJ DLLs.</td>
</tr>
<tr>
<td>prefix.ADSNHFS</td>
<td>This library contains the data that is to be copied into z/OS UNIX System Services.</td>
</tr>
<tr>
<td>prefix.ADSNIVPD</td>
<td>This library contains the IVP input data and expected output for sample applications.</td>
</tr>
<tr>
<td>prefix.ADSNMACS</td>
<td>This library contains the Db2 macros, sample programs, sample data, initialization data, TSO CLISTS, ISPF panels, and ISPF messages.</td>
</tr>
<tr>
<td>prefix.ADXRLOAD</td>
<td>This library contains an individual object module for every IRLM load module.</td>
</tr>
<tr>
<td>prefix.ADXRSAMP</td>
<td>This library contains the installation procedures for installing IRLM Version 2.</td>
</tr>
</tbody>
</table>

The following table describes all Db2 target libraries.

Table 6. Db2 target libraries

<table>
<thead>
<tr>
<th>Target libraries</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix.SDSNBASE</td>
<td>This library contains all jobs that are required to complete SMP/E installation.</td>
</tr>
<tr>
<td>prefix.SDSNC.H</td>
<td>This library contains the header files. The command-line interface (CLI) requires header files. C language application programs can use header files.</td>
</tr>
<tr>
<td>prefix.SDSNCLST</td>
<td>This TSO CLIST library contains code that simplifies the process of installing and migrating, aids program preparation and the use of Db2 utilities, and allows the use of Db2 Interactive (Db2I).</td>
</tr>
<tr>
<td>prefix.SDSNDBRM</td>
<td>This library contains the system DBRMs for Db2 12.</td>
</tr>
<tr>
<td>prefix.SDSNEXIT</td>
<td>This program library is empty when first created. The installation jobs put the DSNZPxxx subsystem parameters load module, application defaults load module (dsnhdcep), and user exit modules into this library.</td>
</tr>
<tr>
<td>prefix.SDSNIVPD</td>
<td>This library contains the IVP input data and the expected output for sample applications.</td>
</tr>
<tr>
<td>prefix.SDSNLINK</td>
<td>This library contains early code of Db2 12.</td>
</tr>
<tr>
<td>prefix.SDSNLOAD</td>
<td>This library contains Db2 12 load modules. This library must be a PDSE data set.</td>
</tr>
<tr>
<td>prefix.SDSNLOD2</td>
<td>This library contains the PDSE data set, which contains JDBC and SQLJ DLLs.</td>
</tr>
</tbody>
</table>
Table 6. Db2 target libraries (continued)

<table>
<thead>
<tr>
<th>Target libraries</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix.SDSNMACS</td>
<td>This macro library contains macros that are needed for the CICS and IMS attachment facilities, the initialization parameter macros, and some data-mapping macros that are needed for some applications.</td>
</tr>
<tr>
<td>prefix.SDSNPFPE or</td>
<td>prefix.SDSNPFPE contains the English task and help panels, and prefix.SDSNPFPPK contains the Kanji task and help panels.</td>
</tr>
<tr>
<td>prefix.SDSNPFPPK</td>
<td></td>
</tr>
<tr>
<td>prefix.SDSNSAMP</td>
<td>This initialization library contains the sample applications and data, the jobs for installing and migrating, the default installation and migration parameters, and catalog initialization data for Db2. The JCLIN for each FMID is stored in this library.</td>
</tr>
<tr>
<td>prefix.SDSNSPFM</td>
<td>This Db2 ISPF message library contains messages that are issued during installation or migrate processing.</td>
</tr>
<tr>
<td>prefix.SDSNSPFPP</td>
<td>This library is the Db2 ISPF library for installation task and help routing panels.</td>
</tr>
<tr>
<td>prefix.SDSNSPFS</td>
<td>This library is the Db2 ISPF skeleton library that is used to produce EDITJCL.</td>
</tr>
<tr>
<td>prefix.SDSNSPFT</td>
<td>This library is the Db2 ISPF command table library.</td>
</tr>
<tr>
<td>prefix.SDXRRESL</td>
<td>This library contains the IRLM load modules. This library might be empty if you chose to install IRLM elsewhere.</td>
</tr>
<tr>
<td>prefix.SDXRSAMP</td>
<td>The IRLM samples library might be empty if you chose to install IRLM elsewhere.</td>
</tr>
</tbody>
</table>

For a description of each SMP/E job that Db2 provides, see the following table. These jobs allocate the Db2 libraries and load them with the data from the installation tapes or cartridges.

Table 7. List of SMP/E jobs

<table>
<thead>
<tr>
<th>Job name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIJAA</td>
<td>This job creates the Db2 target and distribution zones, and defines the SMP/E control data sets in these zones and in the SMP/E global zone.</td>
</tr>
<tr>
<td>DSNACEP1</td>
<td>This job invokes SMP/E to accept all the required and additional FMIDs into the Db2 distribution libraries (DLIBs).</td>
</tr>
<tr>
<td>DSNALLOC</td>
<td>This is the SMP/E allocation job. It creates the Db2 target and distribution libraries and defines the libraries in the SMP/E target and distribution zones for Db2 for the required and optional FMIDs.</td>
</tr>
<tr>
<td>DSNISMKD</td>
<td>This sample job invokes the supplied DSNMKDIR EXEC to allocate HFS paths for JDBC and SQLJ.</td>
</tr>
<tr>
<td>DSNLSMKD</td>
<td>This sample job invokes the supplied DSNLMKDIR EXEC to allocate HFS paths for MQListener.</td>
</tr>
<tr>
<td>DSNWSMKD</td>
<td>This sample job invokes the supplied DSNWMKDIR EXEC to allocate HFS paths for Web Services Object Runtime (WORF).</td>
</tr>
<tr>
<td>DSNAPPL1</td>
<td>This job invokes SMP/E to apply all the FMIDs to the Db2 target libraries for the required and additional FMIDs.</td>
</tr>
<tr>
<td>DSNRECV1</td>
<td>This job invokes SMP/E to receive all the required FMIDs (from both tapes or cartridges) from the base tape into the SMP/E control data sets.</td>
</tr>
<tr>
<td>Job name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>DSNRECV2</td>
<td>This job invokes SMP/E to receive all the required FMIDs (from both tapes or cartridges) for IRLM into the SMP/E control data sets.</td>
</tr>
<tr>
<td>DSNRECV3</td>
<td>This job invokes SMP/E to receive all the additional FMIDs (from both tapes or cartridges) for ODBC, JDBC, and SQLJ into the SMP/E control data sets.</td>
</tr>
<tr>
<td>DSNRECV4</td>
<td>This job invokes SMP/E to receive FMIDs (from both tape or cartridge) for the Kanji DB2I panels into the SMP/E control data sets.</td>
</tr>
<tr>
<td>DSNTIJUD</td>
<td>This job invokes SMP/E to delete all Db2 11 entries from the SMP/E libraries.</td>
</tr>
<tr>
<td>DSNDDEF1</td>
<td>This job creates DDDEF entries for the SMP/E target and distribution libraries.</td>
</tr>
</tbody>
</table>
Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST

The installation CLIST displays a series of ISPF panels that prompt you to supply parameter values or accept the supplied default values. In most cases, the fields on the ISPF panels correspond with Db2 subsystem parameters.

Before you begin

You must use SMP/E to load the required Db2 libraries, as described in Loading the Db2 libraries. Running the installation CLIST requires the following Db2 load libraries:

- `prefix.SDSNCLST`
- `prefix.SDSSPFM`
- `prefix.SDSNSPFP`
- `prefix.SDNSAMP`
- `prefix.SDSNIVPD` (only if generating z/OSMF artifacts)

About this task

The installation CLIST allocates several data sets for input and output. From your TSO user ID, you should be able to allocate these data sets to the permanent or temporary unit names that are provided on installation panel DSNTIPA2. These devices can be defined by an esoteric device group.

The values that you enter or accept on the ISPF panels describe the operating characteristics of your Db2 system. If you change the default values when you install Db2 or when you migrate to a newer release, the CLIST verifies that the values that you enter are within the allowable ranges.

The values that you specify configure the subsystem parameter module and application defaults load module for the Db2 subsystem.

Subsystem parameter module

The subsystem parameter module is generated by job DSNTIJUZ each time that you install, migrate, or update Db2. Six macros expand to form this data-only subsystem parameter load module. The load module contains the Db2 execution-time parameters that you selected using the ISPF panels. These six macros are DSN6ARVP, DSN6FAC, DSN6LOGP, DSN6PRM, DSN6SYSP, and DSN6GRP. By default, this load module has the name that you specify in the PARAMETER MODULE field in panel DSNTIPO3. For more information, see the PARAMETER MODULE field.

Application defaults load module

The application defaults load module, sometimes called the “DSNHDCEP”, is generated by job DSNTIJUA. This module contains the application programming defaults. By default, this load module has the name that you specify in the APPL DEFaulTs field of installation panel DSNTIPO3. For more information, see the APPL DEFAULTS field.
Running the installation CLIST also prepares the jobs that are required for installation or migration. The CLIST tailors each installation or migration job according to the values that you specify on the ISPF panels.

After a successful installation or migration, you can also update these values later to improve your operations. For more information, see Updating subsystem parameter and application default values.

Tip: You can capture most settings of an existing Db2 subsystem by running the DSNTIJXZ job, which invokes the DSNTXAZP tool. Running this job reduces the effort to manually compare settings in the current DSNTIJUZ job with the values that are defined in the defaults input member. For details, see Updating the CLIST defaults input member: job DSNTIJXZ and DSNTXAZP tool.

Important: If you use the installation CLIST, do not change any members in your prefix.SDSNSAMP library. If you do, the installation CLIST can fail or produce unpredictable results when tailoring the installation or migration jobs.

Procedure

To tailor Db2 12 installation and migration jobs, complete the following steps:

1. Make the Db2 ISPF libraries available to TSO by taking one of the following actions:
   - To prepare for running only the Db2 installation CLIST, issue the following commands:
     ```plaintext
     PROFILE WTP MSGID
     ALLOCATE DNAME(SYSEXEC) DSN('prefix.SDNSCLST') SHR REUSE
     ALLOCATE DNAME(ISPMLIB) DSN('prefix.SDNSPFM' +
     'ISP.SISPMENU') SHR REUSE
     ALLOCATE DNAME(ISPPLIB) DSN('prefix.SDNSPFP' +
     'ISP.SISPPENU') SHR REUSE
     ```
   - To prepare for running the Db2 installation CLIST and the Db2 Interactive (DB2I) facility, issue the following commands. If you use Kanji DB2I panels, concatenate SDSNPFPK to ISPPLIB instead of SDSNPFPE.
     ```plaintext
     PROFILE WTP MSGID
     ALLOCATE DNAME(SYSPROC) DSN('prefix.SDNSCLST') SHR REUSE
     ALLOCATE DNAME(ISPMLIB) DSN('prefix.SDNSPFM' +
     'ISP.SISPMENU') SHR REUSE
     ALLOCATE DNAME(ISPPLIB) DSN('prefix.SDNSPFP' 'prefix.SDNSPFPE' +
     'ISP.SISPPENU') SHR REUSE
     ALLOCATE DNAME(ISPSLIB) DSN('prefix.SDNSPFS' +
     'ISP.SISPSLIB' 'ISP.SISPSENU') SHR REUSE
     ```
   
   The PROFILE command provides complete error messages.

   The ALLOCATE command uses the default names of the libraries that contain the ISPF panels. These ISPF library names might be different at your site. To concatenate or merge existing libraries with them, put the library names in the list of names in parentheses after DSN with the largest block size first. (If two or more libraries have the same block size, you can list either one first.)

2. If you are running the CLIST to migrate to Db2 12 update an existing Db2 12 subsystem, or install a new Db2 12 based on the settings of existing installation, configure and run job DSNTIJXZ to update the defaults input member. For details, see Updating the CLIST defaults input member: job DSNTIJXZ.

3. If your z/OS system has multiple procedure libraries, ensure that the installation jobs access the appropriate set of procedures by specifying the order in which procedure libraries are searched. You can use the following JCLLIB statement to specify the order in which procedure libraries are searched.
The JCLLIB statement must follow the JOB statement and precede the first EXEC statement in the job. You can have Db2 insert this statement in your JCL for you by entering the statement on installation panel DSNTIPY. For more information on the JCLLIB statement, see [JCLLIB statement (MVS JCL Reference)](https://www.ibm.com).

4. Establish system affinity to ensure that the installation jobs run on the z/OS system where the appropriate Db2 subsystem is running. The following methods are examples of several z/OS installation-specific methods for establishing system affinity:

- For JES2 multi-access spool (MAS) systems, use the following JCL statement:
  ```
  /*JOBPARM
  **SYSAFF=cccc
  ```
  Where cccc is the JES2 name. You can specify an asterisk (SYSAFF=*) to indicate that the job should run on the system from which it was submitted.

- For JES3 systems, use the following JCL statement:
  ```
  //MAIN SYSTEM=(main-name)
  ```
  The main-name is the JES3 name.

You can edit the jobs manually, or you can enter the preceding statements on installation panel DSNTIPY and have Db2 insert these statements for you. For more information, see [JES2 Control Statements (MVS JCL Reference)](https://www.ibm.com) and [JES3 Control Statements (MVS JCL Reference)](https://www.ibm.com).

5. Invoke the CLIST:

a. Check that your TSO logon region size is sufficient. Usually 2 MB is enough.

b. Invoke ISPF.

c. Select option 6 on the main ISPF panel.

d. Invoke the installation CLIST by entering one of the following specifications:

   - EXEC 'prefix.SDSNCLST(DSNTINST)'
   - If you want to receive the messages that trace the progress of the CLIST, use the following specification:
     ```
     EXEC 'prefix.SDSNCLST(DSNTINST) ' 'CONTROL(LIST)'
     ```
   - If you want to use the CLIST but skip the verification of output data sets, use the following specification:
     ```
     EXEC 'prefix.SDSNCLST(DSNTINST) ' 'CHKOUTDS(NO)'
     ```

By default, the CLIST verifies the following information when installation panels DSNTIPT and DSNTIPTPA1 are displayed:

- The CLIST verifies that the TEMP CLIST LIBRARY and SAMPLE LIBRARY that you specify on DSNTIPT can be allocated and opened for output.
- If you specify a value for OUTPUT MEMBER NAME on DSNTIPTPA1, the CLIST verifies that the data set prefix.SDSSAMP can be allocated and opened for output.

The CLIST reads a set of default values and displays them on the panels. The values can be the original default values that are supplied by IBM, values that you specified in a previous run of the CLIST, or values that were generated when you ran the DSNTXAZP tool.
6. Use the CLIST panels to verify or adjust the values in the panel fields. For instructions for working with CLIST panels and descriptions of the panels and fields, see “The Db2 installation CLIST panel session” on page 61.

7. On the last panel that displays, CLIST calculations (DSNTIPC1 for installation or DSNTIPC for migration), press Enter to complete the CLIST processing. A series of messages provide details about the CLIST processing:
   a. DSNT478I BEGINNING EDITED DATA SET OUTPUT
   b. If the CLIST detects a problem when it checks the parameter values that you entered, you receive an error or warning message indicating the name of the parameter and the type of problem.
      • If you receive an error message, the CLIST cannot edit the installation or migration jobs properly.
      • If you receive a warning message, check the conditions. A warning message can sometimes be issued even when the conditions are normal or acceptable.
      • If you specify several large numbers in the panels, the CLIST might send a message indicating an overflow in CLIST arithmetic.
   The CLIST displays the Main Panel again. You can proceed through the panels, rechecking or changing parameter values.
   c. If the CLIST does not find any errors, you receive messages on DSNTIPC1 that indicate the amount of required disk storage and virtual storage.
   d. You might also receive some other information messages. For example, if you specified an existing data set for the TEMP CLIST LIBRARY field or the SAMPLE LIBRARY field on panel DSNTIPT, the following message is displayed:
      WARNING(S) ISSUED

Results

The installation CLIST tailors each installation or migration job according to the panel values that you specified. For a list of the tailored jobs, see “Installation and migration jobs that the CLIST tailors.” These jobs and other output from the CLIST panel session are stored in several output data sets. For more information, see CLIST panel session output.

For each edited job, you receive the following message:
DSNT489I CLIST EDITING dname(member), explanation

Important:

If an error occurs when the installation CLIST edits your jobs, you receive the following message:
IKJ52506I TEXT NOT FOUND

Enter END NOSAVE to prevent modification of the original copies of your installation or migration jobs.

You can rerun the CLIST with the trace control parameter set to CONTROL(SYMLIST) to learn what caused the problem. In some cases, specifying CONTROL(LIST) as the trace control parameter may provide enough information for you to find the source of the problem.
After the CLIST finishes tailoring the jobs, it displays the Main Panel (DSNTIP A1) again. If you need to continue your tailoring at another time, conclude this session. Then, when you start a new session, use the value that you specified for OUTPUT MEMBER NAME during this session as the value for INPUT MEMBER NAME during the new session. Enter these values on the Main Panel.

What to do next

You might need to tailor several items in the installation jobs, such as data set names or library names, to meet the requirements of your site. For more information see Tailoring the installation jobs.

Related concepts:
- Subsystem parameters (Introduction to Db2 for z/OS)
- Application programming default values (Introduction to Db2 for z/OS)

Related reference:
- Directory of subsystem parameters, panel fields, and application default values

Updating the CLIST defaults input member: job DSNTIJXZ

Before you run the CLIST to migrate to Db2 12, update an existing Db2 12 subsystem, or install Db2 12 using the settings of an existing Db2 12 subsystem, update your defaults input member.

Before you begin

Ensure that you have configured Resource Recovery Services (RRS).

About this task

The DSNTIDxx defaults input member contains, the subsystem parameter settings, buffer pool settings, and other settings from an existing subsystem. It is used as input during your migration, update, or installation.

You can use job DSNTIJXZ to create an updated defaults input member, DSNTIDxx, which contains most of the settings from your current subsystem.

Running this job reduces the effort of manually comparing settings in your current DSNTIJUZ job with the values that are defined in the defaults input member.

Before running this job, you might need to bind or refresh the package and plan for the DSNTXAZP program. The necessary bind commands are in installation job DSNTIJSG.

Procedure

To update the CLIST defaults input member:

1. Obtain the correct version of the DSNTIJXZ job.
   - If you are preparing to migrate to Db2 12, use the Db2 11 DSNTIJXZ job on Db2 11 to create an updated migration input member for the Db2 12 installation CLIST.
   - If you are preparing to update an existing Db2 12 subsystem or install Db2 12 using the settings of an existing Db2 12 subsystem, use the Db2 12 DSNTIJXZ job to create an updated input member.

2. Configure and run job DSNTIJXZ.
If you are preparing to migrate to Db2 12, specify as input your existing Db2 11 DSNTIDxx member. This member is the one that was created when you installed or migrated to Db2 11. If you do not have this member, specify as input the Db2 11 DSNTIDXA member, or the DSNTIDXB member if you use SAP.

For Db2 data sharing, verify that job DSNTIJXZ runs on the data sharing member that you are preparing to migrate and that the input to job DSNTIJXZ is the DSNTIDxx member for that data sharing member.

If you are preparing to update an existing Db2 12 subsystem or install Db2 12 using the settings of an existing Db2 12 subsystem, specify as input your existing Db2 12 DSNTIDxx member. This member is the one that was created when you installed or migrated to Db2 12. If you do not have this member, specify as input the Db2 12 DSNTIDXA member, or the DSNTIDXB member if you use SAP.

Job DSNTIJXZ uses the DSNTXAZP tool to create a copy of the specified DSNTIDxx member and update it with the current subsystem parameter settings, buffer pool settings, and other installation settings.

**Important:** The DSNTXAZP tool does not update the DSNTIDxx CLIST input member with current settings for the following parameters or panel fields.
Update the values as you complete the CLIST, or modify the values directly in the DSNTIDxx member:

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIPA2</td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>CATALOG ALIAS (VCATALOG)</td>
</tr>
<tr>
<td></td>
<td>DEFINE CATALOG (VCATSTAT)</td>
</tr>
</tbody>
</table>
Table 8. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPA3</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>PERMANENT UNIT NAME</td>
</tr>
<tr>
<td></td>
<td>(VOLSDEV)</td>
</tr>
<tr>
<td></td>
<td>TEMPORARY UNIT NAME</td>
</tr>
<tr>
<td></td>
<td>(TVOLSEVT)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACL1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS1)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACL2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS2)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACL3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS3)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: DATACLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACL6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: MGMTCLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACL7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS7)</td>
</tr>
<tr>
<td><strong>DSNTIPK</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>WORK FILE DB (WKDBNAME)</td>
</tr>
<tr>
<td></td>
<td>SUBGRP ATTACH (SGRPATNM)</td>
</tr>
<tr>
<td>CLIST Panel</td>
<td>Field Name (DSNIDxx entry)</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td><strong>DSNTIPG</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>ROUTINES CREATOR (AUTHID)</td>
</tr>
<tr>
<td></td>
<td>SEC DEF CREATOR (SECDEFID)</td>
</tr>
<tr>
<td></td>
<td>INSTALL SQL ID (INSQLID)</td>
</tr>
<tr>
<td></td>
<td>INSTALL PKG OWNER (INSPKOWN)</td>
</tr>
<tr>
<td></td>
<td>INSTALL GRANTEE(S) (INSGRRLST)</td>
</tr>
<tr>
<td></td>
<td>IVP SQL ID (IVPSQLID)</td>
</tr>
<tr>
<td></td>
<td>IVP PACKAGE OWNER (IVPPKOWN)</td>
</tr>
<tr>
<td></td>
<td>IVP GRANTEE(S) (IVPGRLST)</td>
</tr>
<tr>
<td><strong>DSNTIPG1</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>INSTALL IC PREFIX (INSICPRE)</td>
</tr>
<tr>
<td></td>
<td>INSTALL DS PREFIX (INSDSPRE)</td>
</tr>
<tr>
<td></td>
<td>IC DATA SET UNIT NAME (ICDSDEV)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE EXPRESSION (ICDSTPLT)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE DATA CLASS (ICDSDC)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE MGMT CLASS (ICDSMC)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE STOR CLASS (ICDSSC)</td>
</tr>
<tr>
<td><strong>DSNTPH</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>COPY 1 NAME (BSDSNAM1)</td>
</tr>
<tr>
<td></td>
<td>COPY 2 NAME (BSDSNAM2)</td>
</tr>
<tr>
<td></td>
<td>Active logs: COPY 1 PREFIX (LOGSPRE1)</td>
</tr>
<tr>
<td></td>
<td>Active logs: COPY 2 PREFIX (LOGSPRE2)</td>
</tr>
<tr>
<td></td>
<td>Archive logs: COPY 1 PREFIX (ARCHPRE1)</td>
</tr>
<tr>
<td></td>
<td>Archive logs: COPY 2 PREFIX (ARCHPRE2)</td>
</tr>
<tr>
<td><strong>DSNITPT</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>TEMP CLIST LIBRARY (NEWCL)</td>
</tr>
<tr>
<td></td>
<td>SAMPLE LIBRARY (NEWSAMP)</td>
</tr>
<tr>
<td></td>
<td>CLIST LIBRARY (OUTCL)</td>
</tr>
<tr>
<td></td>
<td>APPLICATION DBRM (DBRMLIB)</td>
</tr>
<tr>
<td></td>
<td>APPLICATION LOAD (RUNLIB)</td>
</tr>
<tr>
<td></td>
<td>DECLARATION LIBRARY (SRCLIB)</td>
</tr>
<tr>
<td></td>
<td>LINK LIST LIBRARY (DSNLINK)</td>
</tr>
<tr>
<td></td>
<td>LOAD LIBRARY (DSNLOAD)</td>
</tr>
<tr>
<td></td>
<td>JDBC AND SQLJ DLLS (DSNLOD2)</td>
</tr>
<tr>
<td></td>
<td>MACRO LIBRARY (DSNMACS)</td>
</tr>
<tr>
<td></td>
<td>LOAD DISTRIBUTION (DSNALOAD)</td>
</tr>
<tr>
<td></td>
<td>EXIT LIBRARY (DSNEXIT)</td>
</tr>
<tr>
<td></td>
<td>DBRM LIBRARY (DSNDBRM)</td>
</tr>
<tr>
<td></td>
<td>IRLM LOAD LIBRARY (DSNIRLM)</td>
</tr>
<tr>
<td></td>
<td>IVP DATA LIBRARY (DSNIVPD)</td>
</tr>
<tr>
<td></td>
<td>INCLUDE LIBRARY (DSNCHDRS)</td>
</tr>
</tbody>
</table>
Table 8. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPU</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY (LERUN)</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY 2 (LERUN2)</td>
</tr>
<tr>
<td></td>
<td>IBM LE LINK EDIT LIB (LELKED)</td>
</tr>
<tr>
<td></td>
<td>IBM LE PRELINK MSG LIB (LEPLMSGL)</td>
</tr>
<tr>
<td></td>
<td>HIGH LEVEL ASSEMBLER LIB (IASMLIB)</td>
</tr>
<tr>
<td></td>
<td>C/CPP COMPILER MODULE (CPROG)</td>
</tr>
<tr>
<td></td>
<td>C/CPP COMPILER LIBRARY (CCOMP)</td>
</tr>
<tr>
<td></td>
<td>C/CPP HEADER LIBRARY (CHDRS)</td>
</tr>
<tr>
<td></td>
<td>C/370 COMPILER MESSAGES (CMSGS)</td>
</tr>
<tr>
<td></td>
<td>C LE PRELINKER PHASE (CPRLK)</td>
</tr>
<tr>
<td></td>
<td>CPP LE PRELINKER PHASE (CPPPRLK)</td>
</tr>
<tr>
<td></td>
<td>CPP CLASS LIB HEADERS (CPPHDRS)</td>
</tr>
<tr>
<td></td>
<td>CPP AUTO CALL LIBRARY (CPPAUTCL)</td>
</tr>
<tr>
<td></td>
<td>CPP CLASS LIBRARY (CPPCLASS)</td>
</tr>
<tr>
<td></td>
<td>CPP PROCEDURE LIBRARY (CPPPROC)</td>
</tr>
<tr>
<td><strong>DSNTIPU1</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>COBOL COMPILER LIBRARY (ICOBCOMP)</td>
</tr>
<tr>
<td></td>
<td>COBOL LE PRELINKER PHASE (ICOBPRLK)</td>
</tr>
<tr>
<td></td>
<td>FORTRAN COMPILER LIBRARY (FORTCOMP)</td>
</tr>
<tr>
<td></td>
<td>FORTRAN LINK EDIT LIB (FORTLINK)</td>
</tr>
<tr>
<td></td>
<td>PL/I COMPILER LIBRARY (PLICOMP)</td>
</tr>
<tr>
<td></td>
<td>PL/I LE PRELINKER PHASE (PLIPRLK)</td>
</tr>
<tr>
<td><strong>DSNTIPW</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>SYSTEM MACLIB (MACLIB)</td>
</tr>
<tr>
<td></td>
<td>SYSTEM PROCEDURES (PROCLIB)</td>
</tr>
<tr>
<td></td>
<td>SORT LIBRARY (SORTLIB)</td>
</tr>
<tr>
<td></td>
<td>IMS RESLIB (RESLIB)</td>
</tr>
<tr>
<td></td>
<td>ISPF ISPLINK MODULE (ISPLMOD)</td>
</tr>
<tr>
<td></td>
<td>GDDM MACLIB (GDDMMACS)</td>
</tr>
<tr>
<td></td>
<td>GDDM LOAD MODULES (GDDMLOAD)</td>
</tr>
<tr>
<td></td>
<td>CICS LOAD LIBRARY (CICSMOD)</td>
</tr>
<tr>
<td></td>
<td>CICS MACRO LIBRARY (CICSMAC)</td>
</tr>
<tr>
<td></td>
<td>CICS COBOL LIBRARY (CICSCOB)</td>
</tr>
<tr>
<td></td>
<td>CICS PL/I LIBRARY (CICSPI)</td>
</tr>
<tr>
<td></td>
<td>CICS EXCI LIBRARY (CICSEXCI)</td>
</tr>
</tbody>
</table>
Table 8. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPD</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>DATABASES (NUMDATAB)</td>
</tr>
<tr>
<td></td>
<td>TABLES (NUMTABLE)</td>
</tr>
<tr>
<td></td>
<td>COLUMNS (NUMCOLUMNS)</td>
</tr>
<tr>
<td></td>
<td>VIEWS (NUMVIEWS)</td>
</tr>
<tr>
<td></td>
<td>TABLE SPACES (NUMTABSP)</td>
</tr>
<tr>
<td></td>
<td>PLANS (NUMPLANS)</td>
</tr>
<tr>
<td></td>
<td>PLAN STATEMENTS (NUMSTMTS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGES (NUMPKGS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGE STATEMENTS (NUMPKSTS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGE LISTS (NUMPLPL)</td>
</tr>
<tr>
<td></td>
<td>EXECUTED STMTS (NUMSTMTE)</td>
</tr>
<tr>
<td><strong>DSNTIPS</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>4K SORT PRIMARY SPACE (S4KSPAC)</td>
</tr>
<tr>
<td></td>
<td>4K SORT WORK TS’S (S4KNOTS)</td>
</tr>
<tr>
<td></td>
<td>4K SORT WORK SEG SIZE (S4KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>32K SORT PRIMARY SPACE (S32KSPAC)</td>
</tr>
<tr>
<td></td>
<td>32K SORT WORK TS’S (S32KNOTS)</td>
</tr>
<tr>
<td></td>
<td>32K SORT WORK SEG SIZE (S32KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP PRIMARY SPACE (T4KSPAC)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP WORK TS’S (T4KNOTS)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP WORK SEG SIZE (T4KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>32K TEMP PRIMARY SPACE (T32KSPAC)</td>
</tr>
<tr>
<td></td>
<td>32K TEMP WORK TS’S (T32KNOTS)</td>
</tr>
<tr>
<td><strong>DSNTIPM</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>SUBSYSTEM NAME (MVSNAME)</td>
</tr>
<tr>
<td></td>
<td>COMMAND PREFIX (MVSPREF)</td>
</tr>
<tr>
<td><strong>DSNTIPR</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>DB2 LOCATION NAME (DDLOC)</td>
</tr>
<tr>
<td></td>
<td>DB2 NETWORK LUNAME (DDLUNAM)</td>
</tr>
<tr>
<td><strong>DSNTIPS</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>DRDA PORT (DRDAPRT)</td>
</tr>
<tr>
<td></td>
<td>RESYNC PORT (RESYNPRT)</td>
</tr>
<tr>
<td><strong>DSNTIPX</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>WLMENV (WLMENV)</td>
</tr>
<tr>
<td><strong>DSNTIPR1</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>All fields (WLMENVx).</td>
</tr>
<tr>
<td><strong>DSNTIPRA</strong></td>
<td>On panels DSNTIPRB - DSNTIPRQ:</td>
</tr>
<tr>
<td></td>
<td>All WLM ENV NAME fields</td>
</tr>
<tr>
<td></td>
<td>(RTxnnWLM)</td>
</tr>
<tr>
<td><strong>DSNTIPY</strong></td>
<td>All numbered job card information fields.</td>
</tr>
</tbody>
</table>

3. When you run the CLIST, specify the DSNTIDxx member that is created by this job during installation or migration:
• For migration, specify the DSNTIDxx member that is created by this job in the DATA SET(MEMBER) NAME field on panel DSNTIPA1.
• During installation, or for updates, specify the DSNTIDxx member that is created by this job in the INPUT MEMBER NAME field on panel DSNTIPA1.

Related reference:
DATA SET(MEMBER) NAME field
INPUT MEMBER NAME field
Directory of subsystem parameters, panel fields, and application default values

DSNTXAZP tool
The DSNTXAZP tool updates a CLIST defaults input member (DSNTIDxx) with certain current subsystem parameter and buffer pool settings that are defined in a specified Db2 subsystem.

Job DSNTIJXZ invokes the DSNTXAZP tool.

The DSNTXAZP tool requires Resource Recovery Services (RRS).

The DSNTXAZP tool does not update all entries in the DSNTIDxx member. For a list of entries that are not updated, see Updating the CLIST defaults input member: job DSNTIJXZ.

The batch program requires the following arguments:
//STEP01 EXEC PGM=DSNTXAZP,PARM="ssid action"

ssid is the subsystem ID.

action is one of the following values:

UPDATE_ALL
Creates a new CLIST defaults input member and updates it with the current subsystem parameter and buffer pool settings.

UPDATE_BPOOL
Creates a new CLIST defaults input member and updates it with the current buffer pool settings only (no subsystem parameter changes).

UPDATE_ZPARAM
Creates a new CLIST defaults input member and updates it with the current subsystem parameter settings only (no buffer pool changes).

The following DD statements are required for processing:

XAINPUT
Used as input to the DSNTXAZP program. It allocates the current CLIST defaults input member that will be used as the source for creating a new CLIST defaults input member. The version of the member should match the version of the selected Db2 subsystem.

XAOUTPUT
Used as output by the DSNTXAZP program. It allocates the generated CLIST defaults input member with updated subsystem parameter and buffer pool settings. This should be a member of a PDS that is defined with RECFM=FB, LRECL=80.

SYSPRINT
Used as output by the DSNTXAZP program for informational and error
messages. It also contains a report on the parameters that were processed and modified. For additional details on the report, see the example below.

The following JCL is an example of running the tool:

```plaintext
//DSNTIXZ EXEC PGM=DSNTXAZP,PARM='DSN UPDATE_ALL'
//XAINPUT DD DSN=prefix.SDSNSAMP(DSNTIDXA),DISP=SHR
//XAOUTPUT DD DSN=prefix.SDSNSAMP(DSNTIDNU),DISP=SHR
//SYSPRINT DD SYSOUT=
```

The tool generates a CLIST input member with the current subsystem parameter settings, buffer pool settings, or both, and the following prolog:

```plaintext
* **********************************************************************
* CREATED MEMBER = MY.DATASET.NAME(DSNTIDNU) *
* *
* CREATED ON = MAR 24, 2010 11:12:30 *
* *
* SOURCE MEMBER = MY.SOURCE.NAME(DSNTIDXA) *
* *
* DESCRIPTIVE NAME = DB2 INSTALLATION DATA *
* *
* FUNCTION = PROVIDE UPDATED VALUES FOR MIGRATING TO A NEW RELEASE *
* OF DB2 OR INSTALLING A NEW RELEASE OF DB2 *
* *
* PARAMETERS:
* *
* PARM NAME = NAME FOR ISPF PANEL VARIABLE *
* PARM TYPE = TYPE FOR VALIDITY CHECKING *
* DS SCP = DATA SHARING SCOPE: G = GROUP, M = MEMBER *
* MIN, MAX = RANGE VALUES FOR VALIDITY CHECKING *
* CURRENT VALUE = VALUE USED AS INPUT FOR NAMED ISPF VARIABLE *
* *
* **********************************************************************
```

In addition to creating a new CLIST input member, the DSNTXAZP tool provides a report of all parameters. The report indicates whether each parameter was updated or if the original value from the input member was retained. It also shows the mapping between the subsystem parameter and the CLIST parameter. This is a sample report:

```
CLIST INPUT MEMBER GENERATION REPORT ** 2010-03-24 08:25:35
-----------------------------------------------------------
CLIST PARAMETER REPORT:

0001 PARAMETER NAME = ABEXP
  ZPARM BUFFERPOOL PARAMETER = ABEXP
  PARAMETER TYPE = CHAR
  DATA SHARING SCOPE = M
  MINIMUM VALUE = NO
  MAXIMUM VALUE = YES
  CURRENT CLIST VALUE = YES
  CURRENT INSTALLED VALUE = YES
  STATUS = RETAINED

0002 PARAMETER NAME = ABIND
  ZPARM BUFFERPOOL PARAMETER = ABIND
  PARAMETER TYPE = CHAR
  DATA SHARING SCOPE = M
  MINIMUM VALUE = NONE
  MAXIMUM VALUE = NONE
  CURRENT CLIST VALUE = YES
```

The Db2 installation CLIST panel session

The installation CLIST displays a series of ISPF panels that prompt you to supply parameter values or accept the supplied default values. In most cases, the fields on the ISPF panels correspond with Db2 subsystem parameters.

For instructions for invoking the Db2 installation and migration CLIST, see Chapter 4, “Tailoring Db2 12 installation and migration jobs with the CLIST,” on page 49.

In general, you can change the default values to match the needs of your organization. However, some values cannot be changed, depending on the INSTALL TYPE value that you specify on panel DSNTIPA1, and whether you are installing the first or subsequent members of a data sharing group.
Navigating in the CLIST panel session

All panel sequences begin with the Main Panel, DSNTIPA1.

You can use the following approaches to navigate in the CLIST panel session:

- To display the panel identifiers, enter the following command in any panel:
  PANELID ON.
- Enter your choices on a panel in the space that is marked by an arrow (==>).
  Begin your entry in the second position to the right of the arrow. The first position is protected and you cannot write in it.
- To validate entries in the current panel and display the next panel, press Enter.
- Panel fields that are marked with asterisks contain values that are primed on the basis of values from a previous panel. The following message is found on these panels:
  DSN444I SCROLLING BACKWARD MAY CHANGE FIELDS MARKED WITH ASTERISKS
  If you scroll back to the panel that has the original value, the values on the succeeding panels are refreshed only if the original value is changed. If the values are changed, the following message is displayed:
  DSN443I VALUES MARKED WITH AN ASTERISK HAVE BEEN UPDATED
  For example, panel DSNTIPH has fields that are marked with asterisks indicating values that are primed on the basis of the CATALOG ALIAS value on installation panel DSNTIPA2.
- To scroll back to the previous panels to review or change values press the End key (usually PF3).
- To save your changes at any point during the panel session, enter the following command in any installation panel: SAVE Then select option 1 in panel DSNTIPSV.
The changes are saved in the output member that is specified in the OUTPUT MEMBER NAME field on panel DSNTIPA1.

- To exit completely from the installation process, press the Return key (usually PF4).

The CLIST session completes when you press Enter from the last panel: DSNTIPC1 for installation or DSNTIPC for migration. For more information about completing the CLIST panel session, see step 7 in Chapter 4, “Tailoring Db2 12 installation and migration jobs with the CLIST,” on page 49.

CLIST panel and field descriptions

The CLIST panel field descriptions following information:

Acceptable values

This part of a panel field description indicates the range or list of allowable values for an installation panel field. The acceptable values apply to the installation panel field, but not necessarily to the subsystem parameter that is associated with that field. Differences might occur if a panel field is used to determine the value of the associated subsystem parameter and the values do not correspond exactly.

In most cases, if a maximum supported panel field value is large enough, you can specify the following multipliers:

- valueK for value multiplied by 1024
- valueM for value multiplied by 1,048,576
- valueG for values greater than 1024 MB
- valueT for values greater than 1024 GB

The maximum acceptable values might be too large for smaller systems; therefore, ensure that the values that you enter are valid for the size of your system.

Default values

The default values that documentation shows are the original values that are supplied by IBM. If you previously ran the CLIST and saved the updated panel values in a DSNTIDxx input data set, the previously entered values appear as default values on the panels.

Tip: You can capture most settings of an existing Db2 subsystem by running the DSNTIJXZ job, which invokes the DSNTXAZP tool. Running this job reduces the effort to manually compare settings in the current DSNTIIZU job with the values that are defined in the defaults input member. For details, see Updating the CLIST defaults input member: job DSNTIJXZ and DSNTXAZP tool.
Panel values that are modified outside of the installation process are not saved in the DSNTIDxx data set and are not reflected on the panels.

**Update**

This information identifies a corresponding field on the Update Selection Panel (DSNTIPB) or refers to information that provides instructions for updating the field value. For more information, see [DSNTIPB: Update selection menu panel](#).

**dsnhdecp**

These are the names of the parameters in the application defaults load module dsnhdecp. For more information, see [Application defaults parameters](#).

**DSNZPxxx**

These are the names of the parameters in the data-only load module DSNZPxxx.

Some Db2 subsystem parameters do not appear on CLIST panels. For details, see [Subsystem parameters that are not on installation panels](#).

**DSNTIPA1 or DSNTILA1: Main panel**

The entries on the Main panel control input to and output from the installation CLIST.

When processing is complete, this panel is displayed again. The values that you enter are saved in the ISPF profile for your authorization ID and are displayed each time that you run the CLIST.

To save your panel input, you must specify an output member name in the OUTPUT MEMBER NAME field.

The DSNTINST CLIST saves the panel input into your DSNTIDxx output member just before the CLIST issues this message:

```
DSNT4781 BEGINNING EDITED DATA SET OUTPUT
```
Recommended approach for new installers

If you are installing for the first time, try the following suggestions.

- In the INSTALL TYPE field, enter INSTALL.
- Set the PREFIX and SUFFIX fields to the values that you used when you allocated the Db2 libraries by using job DSNALLOC.
- In the INPUT MEMBER NAME field, use DSNTIDXA (the default) for the first run. If you are using SAP, specify DSNTIDXB, the SAP-specific CLIST input member, instead. For any later runs, the CLIST sets the default input name to the prior output name.
- Specify a value in the OUTPUT MEMBER NAME field to save your options.
- Specify values in the TEMP CLIST LIBRARY, CLIST LIBRARY, and SAMPLE LIBRARY fields on installation panel [DINSTIPT] when you want output data sets tailored.

**Important:** Do not run the installation jobs before tailoring them with the CLIST. If you want to update the parameters later, the CLIST sets the default input name to the prior output name.

**DSNTILA1: Main panel for generating z/OSMF workflows and provisioning Db2 subsystems**

When you specify USE Z/OSMF WORKFLOW ==> YES, panel DSNTILA1 opens, and the CLIST runs with a modified set of installation panels that support the generation of z/OSMF workflows. For more information about the modified panels, see [Installation panels for installing or migrating with z/OSMF](#).

You can also specify INSTALL TYPE ==> PROVISION to specify that you are running the CLIST to prepare artifacts for services that provision Db2 subsystems. For more information, see [Creating artifacts for provisioning Db2 subsystems by running the installation CLIST](#).
The CLIST continues to run starting with the DSNTILA1 and the other modified panels unless and until you set USE Z/OSMF WORKFLOW ==> NO.

**Related tasks:**
- Tailoring Db2 12 installation and migration jobs with the CLIST
- Automating Db2 installation or migration with z/OS Management Facility
- Activating Db2 12 function levels (Db2 for z/OS What's New?)

**Related reference:**
- What's new in Db2 12 function levels (Db2 for z/OS What's New?)

**USE Z/OSMF WORKFLOW field**
The USE Z/OSMF WORKFLOW field specifies whether to run the CLIST with panels that support the creation of z/OSMF workflows for migrating or installing Db2.

Acceptable values: NO, YES
Default: NO
DSNHDECOP: None

**NO**
The CLIST uses a set of installation panels that support tailoring the CLIST in preparation for installing or migrating Db2, as described in Tailoring Db2 12 installation and migration jobs with the CLIST.

**YES**
The CLIST uses a modified set of panels that support the generation z/OSMF workflow definition files and input variable files.

As soon as you set USE Z/OSMF WORKFLOW ==> YES, panel DSNTILA1 opens. The CLIST then runs starting with the DSNTILA1 panel each time, until you set USE Z/OSMF WORKFLOW ==> NO. Then, the CLIST runs starting with panel DSNTIPA1 again.

**Related tasks:**
- Automating Db2 installation or migration with z/OS Management Facility
Related reference:

Installation panels for installing or migrating with z/OSMF

INSTALL TYPE field
The value of the INSTALL TYPE field specifies whether you are installing,
updating, or migrating to Db2 12, activating a Db2 12 function level, or preparing
artifacts for creating services to provision Db2 subsystems.

Acceptable values: ACTIVATE, INSTALL, UPDATE, MIGRATE, PROVISION
Default: INSTALL
DSNHDECP: none

INSTALL
Specifies that you are installing Db2 for the first time. This is the default value
for the first run of the CLIST. After you complete the installation, function
level 500 is activated.

MIGRATE
Specifies that you are migrating from Db2 11 to Db2 12.
When you are migrating, the DATA SET(MEMBER) NAME field is required.

ACTIVATE
Specifies that you are activating a Db2 function level. When this option is
specified, Db2 prompts you to specify the new function level on the DSNTIP00
panel. In a data sharing environment, the function level activation process is
completed only once for the data sharing group. The CLIST customizes a
minimal job set for activation of the new function level.

Important: Use this option to activate a higher function level only after
migration to Db2 12 is complete and Db2 is stable at a particular function
level.

UPDATE
Specifies that you are updating parameters for an existing Db2 subsystem.

PROVISION
Specifies that you are preparing artifacts for creating services to provision Db2
subsystems. When you specify this value panel DSNTIPAP is displayed next in
the CLIST panel session. For more information, see Creating artifacts for
provisioning Db2 subsystems by running the installation CLIST.

If you are updating or migrating, you use the same set of panels that you use for
installation. Each panel displays all fields; however, the fields that cannot be
changed in update or migrate mode are protected. This way, you can see the
values that are related to ones that you want to change.

You can also choose either INSTALL or UPDATE to recheck values you that chose
before.

Certain fields cannot be changed during a migration. See panels DSNTIPH,
DSNTIPD, DSNTIP7, and DSNTIPP for more information. Ensure that those fields
are correct in the data set member that you provide.

Related tasks:
Installing Db2
Updating subsystem parameter and application default values

Related reference:
**What's new in Db2 12 function levels (Db2 for z/OS What's New?)**

**DATA SHARING field (DSHARE subsystem parameter)**

The DSHARE subsystem parameter controls whether the data sharing function is to be used.

Acceptable values: YES, NO, blank (for update)
Default: NO
DSNZPxxx: DSN6GRP DSHARE

Choose NO if you are not using data sharing. If you choose YES, you will continue to panel DSNTIPK after completing panel DSNTIPA2.

**Installation**

If you specify YES, this window is displayed:

```
DSNTIPA

DATA SHARING FUNCTION:
Select one.
  1. Group
  2. Member
  3. Enable

PRESS: ENTER to continue
RETURN to exit
```

*Figure 4. DSNTIPA*

**DATA SHARING FUNCTION:**

Specify a data sharing function.

Acceptable values: Group, Member, Enable
Default: none
DSNZPxxx: none

A value is required. After entering a value, you proceed to panels DSNTIPA2 and DSNTIPK.

**Migration**

If you specify YES in the DATA SHARING field, a window is displayed asking if the current member is the first to migrate.
**FIRST MEMBER OF GROUP TO MIGRATE?**

Specify Yes if this is the first member of a data sharing group to migrate.

Acceptable values: Yes, No
Default: none
DSNZPxxx: none

A value is required. After entering a value, you proceed to panels DSNTIPA2 and DSNTIPK.

**Related tasks:**
- Adding a new data sharing member
- Enabling Db2 data sharing
- Installing a new data sharing group

**DATA SET(MEMBER) NAME field**

The value of the DATA SET(MEMBER) NAME field indicates the name of the input data set to use for migrating from Db2 11. If you are installing, updating, or provisioning Db2 subsystems, the DATA SET(MEMBER) NAME field must remain blank. If you are installing or updating, the field must remain blank.

Acceptable values: 1 - 44 alphanumeric characters
Default: NULL
DSNZPxxx: none

The named member contains the output parameters that were produced when you last installed, updated, or migrated the prior release of Db2. Provide the fully qualified data set name as shown in the following example:

DSN1110.SDSNSAMP(DSNTIDX)

If you no longer have this data set member, or if the one you have is incorrect, use the installation or update process from your previous release to re-create or correct the member. Enter the correct values on the panels, and save them under a new output member name. Discard the JCL that is created by this process; use the newly created member for migration.

Alternatively, you can run job DSNTIJXZ on your subsystem to create an updated data set member that contains the subsystem parameter and buffer pool settings from your previous release.
Valid data set name

Valid data set names can be unqualified or qualified:

Unqualified name
One to eight alphanumeric or national characters, a hyphen, or the character X'C0'. The first character must be alphabetic or national. Do not use hyphens in data set names for RACF-protected data sets. For example, ALPHA is an unqualified data set name.

Qualified name
Multiple names joined by periods. Each name is coded like an unqualified name. Therefore, the name must contain a period within every eight characters. For example, ALPHA.PGM is a qualified data set name. The maximum length of a qualified data set name is:
- 44 characters if you use the TSO PROFILE setting NOPREFIX.
- 42 characters if you use the TSO PROFILE setting PREFIX.
- For an output tape data set, 17 characters, including periods. If the name is longer than 17 characters, only the right-most 17 characters are written to the tape header label.

Related tasks:
- Updating the CLIST defaults input member: job DSNTIJXZ
- Migrating an existing data sharing group to Db2 12

LIBRARY NAME PREFIX field
The LIBRARY NAME PREFIX field specifies the input prefix for the SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNDBRM, and SDSNCLST libraries.

Acceptable values: 1 - 18 characters
Default: DSN1210
DSNZPxxx: none

The prefix must be the same as the name that you specified for the symbolic parameter TARGPRE in SMP/E job DSNALLOC.

This is the prefix of the Db2 SMP/E target libraries. If you specify DSNTIDXA or DSNTIDXB in the INPUT MEMBER NAME field, the prefix for the INCLUDE LIBRARY field, and the prefix and suffix for the LINK LIST LIBRARY field through the IVP DATA LIBRARY field on "DSNTIPT: Db2 data set names panel" on page 114 are set. If all these data sets do not have the same prefix and suffix, you can change them on installation panel DSNTIPT.

Related reference:
- INPUT MEMBER NAME field

LIBRARY NAME SUFFIX field
The LIBRARY NAME SUFFIX field specifies a suffix for the names of the libraries that are to be used by installation and sample jobs.

Acceptable values: 1 - 17 characters
Default: NULL
DSNZPxxx: none

The fully qualified data set name cannot exceed 44 characters. Names that exceed eight characters must be in groups of no more than eight characters, separated by periods.
Use a suffix only if you have added a common suffix to the following libraries when you created them in job DSNTIJAE:

- prefix.ADSNLOAD
- prefix.ADSNEXIT
- prefix.ADSNMACS
- prefix.SDSNLOAD
- prefix.SDSNEXIT
- prefix.ADSNMACS
- prefix.SDSNCLST
- prefix.SDSNSAMP
- prefix.SDSNDBRM
- prefix.SDXRRESL
- prefix.SDSNIVPD

If you did not add a common suffix to these libraries, enter their correct data set names on panel DSNTIPT.

To use the default Db2 data set names, specify DSN1010 in the LIBRARY NAME PREFIX field, and leave the LIBRARY NAME SUFFIX field blank.

**DATA SET NAME PREFIX field**

The DATA SET NAME PREFIX field specifies the input prefix for the Db2 sample application libraries and for data sets that are to be created by the CLIST.

Acceptable values: 1 - 18 characters that form a valid z/OS data set name prefix

Default: DSN1210

DSNZPxxx: none

The prefix applies to the following data sets. The fully qualified data set name cannot exceed 44 characters. Names that exceed eight characters must be in groups of no more than eight characters that are separated by periods.

- prefix.NEW.SDSNTEMP
- prefix.NEW.SDSNSAMP
- prefix.NEW.SDSNCLST
- prefix.DBRMLIB.DATA
- prefix.RUNLIB.LOAD
- prefix.SRCLIB.DATA

If you specified YES in the USE Z/OSMF WORKFLOW field on panel DSNTIPA1, the installation CLIST creates two additional libraries to which the prefix applies:

- prefix.NEW.SDSNMFSA
- prefix.NEW.SDSNSAM2

These data sets are partitioned data sets. The first two are deleted if they exist, and they are created or re-created during the tailoring session. If you specify DSNTIDXA or DSNTIDXB in the INPUT MEMBER NAME field, the prefix for fields 1, 2, and 3, and the prefix and suffix for fields 4, 5, and 6 on installation panel **DSNTIPT: Db2 data set names panel** are set. If all these data sets do not have the same prefix and suffix, you can change them on installation panel DSNTIPT.

**DATA SET NAME SUFFIX field**

The DATA SET NAME SUFFIX field specifies the suffix for the Db2 sample application libraries and for data sets that are to be created by the CLIST.

Acceptable values: 1 to 17 characters that form a valid z/OS data set name prefix

Default: blank

DSNZPxxx: none
The suffix applies to the following data sets. The fully qualified data set name cannot exceed 44 characters. Names that exceed eight characters must be in groups of no more than eight characters that are separated by periods.

- prefix.DBRMLIB.DATA
- prefix.RUNLIB.LOAD
- prefix.SRCLIB.DATA

If you do not use a common suffix to these libraries, enter their correct data set names on panel DSNTIPT.

To use the default Db2 data set names, specify DSN1110 in the DATA SET NAME PREFIX field, and leave the DATA SET NAME SUFFIX field blank.

**INPUT MEMBER NAME field**

The INPUT MEMBER NAME field specifies the member name of the input data set. This data set contains the default Db2 12 parameter values that are used for installation and migration, as in `prefix.SDSNSAMP.suffix`.

Acceptable values: 1 - 8 characters
Default: DSNTIDX
DSNZPxxx: none

**Installing Db2**

If you are installing Db2 for the first time and not using SAP, use the supplied defaults input member, DSNTIDX. If you are using SAP, you should specify DSNTIDX, the SAP-specific input member. If you process the panels several times within a single run of the CLIST, all the previous values that are entered, except edited output data sets, remain the same.

To install Db2 using default parameter values from a previous installation, you must supply the member that contains the output from the previous run of the CLIST. That member was specified in the OUTPUT MEMBER NAME field during the previous run of the CLIST.

**Recommendation:** Before rerunning the CLIST, use job DSNTIJXZ to refresh the output member from the previous run of the CLIST.

**Migrating to Db2 12**

If you are migrating, you must specify two member names for input values: one in the INPUT MEMBER NAME field, and one in the DATA SET(MEMBER) NAME field. The INPUT MEMBER NAME field must specify a member that contains the default parameter values for the new release (usually DSNTIDX, or DSNTIDX if you use SAP). This member is applied first to establish the CLIST parameters.

The member that is specified in the DATA SET(MEMBER) NAME field must be different from the one that is specified in the INPUT MEMBER NAME field. The DATA SET(MEMBER) NAME field must specify a member that contains the Db2 11 values at your site. You can create this member by running job DSNTIJXZ on your previous release. This member is applied last and it overrides the CLIST values that are established by the member that is specified in the INPUT MEMBER NAME field.
Updating an existing Db2 subsystem

Before running the CLIST in UPDATE mode, use job DSNTIJXZ to refresh the output member from the previous run of the CLIST. Specify the refreshed member in the INPUT MEMBER NAME field.

The following table lists the data set names that are generated with the prefix and suffix values from PREFIX and SUFFIX only when the INPUT MEMBER NAME is DSNTIDXA or DSNTIDXB.

<table>
<thead>
<tr>
<th>Default library name</th>
<th>CLIST edited library name</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix.DBRMLIB.DATA</td>
<td>prefix.DBRMLIB.DATAsuffix</td>
</tr>
<tr>
<td>prefix.RUNLIB.LOAD</td>
<td>prefix.RUNLIB.LOAD.suffix</td>
</tr>
<tr>
<td>prefix.SRCLIB.DATA</td>
<td>prefix.SRCLIB.DATA.suffix</td>
</tr>
<tr>
<td>prefix.SDSNDNDBRM</td>
<td>prefix.SDSNDNDBRM.suffix</td>
</tr>
<tr>
<td>prefix.SDNSLINK</td>
<td>prefix.SDNSLINK.suffix</td>
</tr>
<tr>
<td>prefix.SDNSLOAD</td>
<td>prefix.SDNSLOAD.suffix</td>
</tr>
<tr>
<td>prefix.SDSNMACS</td>
<td>prefix.SDSNMACS.suffix</td>
</tr>
<tr>
<td>prefix.ADSNLOAD</td>
<td>prefix.ADSNLOAD.suffix</td>
</tr>
<tr>
<td>prefix.ADSNMACS</td>
<td>prefix.ADSNMACS.suffix</td>
</tr>
<tr>
<td>prefix.SDNSAMP</td>
<td>prefix.SDNSAMP.suffix</td>
</tr>
<tr>
<td>prefix.SDNSCLST</td>
<td>prefix.SDNSCLST.suffix</td>
</tr>
<tr>
<td>prefix.SDNSIVPD</td>
<td>prefix.SDNSIVPD.suffix</td>
</tr>
<tr>
<td>prefix.SDNSC.H</td>
<td>prefix.SDNSC.H</td>
</tr>
<tr>
<td>prefix.SDXRRESL</td>
<td>prefix.SDXRRESL.suffix</td>
</tr>
</tbody>
</table>

Related tasks:
- Adding a new data sharing member
- Migrating an existing data sharing group to Db2 12

Related reference:
- DATA SET(MEMBER) NAME field

OUTPUT MEMBER NAME field

The OUTPUT MEMBER NAME field specifies the member name of the output data set in which to save the values that are entered on the installation panels.

Acceptable values: 1 - 8 characters
Default: NULL
DSNZPxxx: none

If you do not specify a name, the values are lost when you leave the installation CLIST, and you no longer have the values available for future updates. This member is stored in prefix.SDNSAMP (not the one created by the DSNTINST CLIST). To avoid replacing any members of prefix.SDNSAMP that were shipped with the product, specify DSNTIDxx as the value of OUTPUT MEMBER NAME, where xx is any alphanumeric value except SA, VB, XA, or XB.

Always give a new value in the OUTPUT MEMBER NAME field for a new panel session. You supply the name from your current session in the INPUT MEMBER NAME field for your next session. You should not use the same member name for output as for input.
Recommendation: Write down the output member name that you entered for reference during future sessions.

**DSNTIPO1: OTC license usage panel**

Use the DSNTIPO1 panel to indicate whether Db2 is to operate under a one-time license charge. This panel is displayed only if you purchased Db2 12 for z/OS Value Unit Edition (FMID JDBCC1Z) for use with z/OS on zNALC.

---

On this panel, specify whether this Db2 subsystem or data sharing member is to be configured to operate using the OTC license. Enter a 1 (NO) or 2 (YES).

**DSNTIPO2: Notice and acceptance of OTC license panel**

Panel DSNTIPO2 is displayed only when you indicate on panel DSNTIPO1 (OTC license usage) that an OTC license is or will be in effect on this Db2 subsystem or data sharing member.

This panel displays the terms of that license and checks for your consent to abide by them. Review the terms by using the ISPF Up and Down keys to scroll through the text.

---

Figure 6. OTC license usage panel: DSNTIPO1

---

Figure 7. Notice and acceptance of OTC license panel: DSNTIPO2
LICENSE TERMS ACCEPTED field (OTC_LICENSE subsystem parameter)

The OTC_LICENSE subsystem parameter indicates whether you accept the terms of the one-time charge (OTC) license.

Acceptable values: YES
Default: none
DSNZPxxx: DSN6SYSPOTC_LICENSE

This field accepts only one value, YES. You cannot pass this panel with any other value in that field. If you do not accept the terms of this license, press the Return key to exit the Db2 installation CLIST.

DSNTIPAP: Provision DB2 - provisioning preferences and data set names

The fields on panel DSNTIPAP specify preferences for the provisioning of Db2, and the names of Db2 libraries that are used for provisioning Db2 subsystems only (not in installation or migration of Db2 12).

Panel DSNTIPAP is only displayed if INSTALL TYPE ===> PROVISION is specified on panel DSNTIPA1 or DSNTILA1. It is displayed immediately after panel DSNTILA1 and before panel DSNTIPA2.

```
SOFTWARE SERVICE INSTANCE NAME PREFIX
Specifies the first two characters of the software service instance name. Must not be blank.
Acceptable values: 2 alphabetic characters (A-Z); the first must not be 'I'.
Default: DY
Update: none
DSNZPxxx: none
z/OSMF variable: none
```

Data sets allocated by SMP/E and other methods:

```
s 4 ENGLISH TASK AND PANELS LIBRARY
    ===> prefix.SDSNPF.FE.suffix
s 5 ISPF MESSAGE LIBRARY PANELS LIBRARY
    ===> prefix.SDSNSPPM.FE.suffix
s 6 ISPF INSTALL TASK AND PANELS LIBRARY
    ===> prefix.SDSNSPP.FE.suffix
s 7 ISPF SKELETON LIBRARY
    ===> prefix.SDSNSPF.SF.suffix
s 8 ISPF COMMAND TABLE PANELS LIBRARY
    ===> prefix.SDSNSPFT.FE.suffix
```

PANELS LIBRARY

PRESS: ENTER to continue RETURN to exit HELP for more information
DB2 COMMAND PREFIX FIRST CHARACTER

Specifies the first character of the generated command prefix for the provisioned Db2 subsystem. Must not be blank.

Acceptable values: One non-alphanumeric character; it must be a character from the table in COMMAND PREFIX field except for the ampersand (&).

Default: -
Update: none
DSNZPxxx: none
z/OSMF AGSSIDPX

Note:
1. The CLIST does not support use of the ampersand (&). However, to use & for a provisioned Db2 subsystem, you can edit dsntivin directly in zFS.

DATA SET OR LIBRARY QUALIFIER

Specifies the intermediate qualifier for the data set or library for customized copies of Db2 target library members during provisioning.

It is NEW in the generated library name prefix.NEW.ssin.SDSNCLST for the CLIST LIBRARY field on panel DSNTP1, where:
• prefix is the INSTALL DS PREFIX from panel DSNTIPG1 (INSTALLATION PREFERENCES PANEL 2)
• ssin is the provisioning software service instance name

Must not be blank.

Acceptable values: Valid data set name qualifier.
Default: NEW
Update: none
DSNZPxxx: none
z/OSMF AGNEWIDS

ENGLISH TASK AND PANELS LIBRARY

Specifies the English task and help panels library. Must not be blank.

Acceptable values: Valid data set name.
Default: prefix.SDSNPFPE.suffix
Update: none
DSNZPxxx: none
z/OSMF DSNPFPE

ISPF MESSAGE LIBRARY

Specifies the Db2 ISPF message library which contains messages that are issued during installation or migrate processing. Must not be blank.

Acceptable values: Valid data set name.
Default: prefix.SDSNSPFM.suffix
Update: none
DSNZPxxx: none
z/OSMF DSNSPFM

ISPF INSTALL TASK AND PANELS LIBRARY

Specifies the Db2 ISPF library for installation task and help routing panels. Must not be blank.
ISPF SKELETON LIBRARY

Specifies the Db2 ISPF skeleton library that is used to produce EDITJCL. Must not be blank.

Acceptable values: Valid data set name.
Default: prefix.SDSNSPFP.suffix
Update: none
DSNZPxxx: none
z/OSMF DSNSPFP

ISPF COMMAND TABLE LIBRARY

Specifies the Db2 ISPF command table library. Must not be blank.

Acceptable values: Valid data set name.
Default: prefix.SDSNSPFS.suffix
Update: none
DSNZPxxx: none
z/OSMF DSNSPFS

Related tasks:

Creating artifacts for provisioning Db2 subsystems by running the installation CLIST
Creating services for provisioning Db2 for z/OS subsystems (DBaaS)

Related reference:

DSNTIPA1 or DSNTILA1: Main panel
INSTALL TYPE field

DSNTIPA2: Db2 catalog and directory panel

The entries on the DSNTIPA2 panel define the name of the ICF catalog for Db2 VSAM data sets and indicate whether to create the specified catalog. The entries also define z/OS Storage Management Subsystem (SMS) classes that are to be used when defining data sets for the Db2 catalog and directory.

The values that you enter on this and each of the subsequent panels are saved in the data set member that you named in the OUTPUT MEMBER NAME field on the Main Panel, DSNTIPA1.

Db2 12 requires SMS, so ensure that the following conditions are met before you proceed:

• SMS is active on the system where this Db2 subsystem resides or is being installed or migrated. SMS must be configured to allocate all data sets for the Db2 catalog and directory in extended format and to use extended addressability.

• Your system storage administrator has provided the required SMS automatic class selection (ACS) routines for data sets and Db2 storage groups created during Db2 installation or migration.
Your system storage administrator has provided any SMS data classes, management classes, and storage classes that you plan to specify on this panel and the DSNTIPA3 panel.

To help performance and recovery, reduce dependency on a single storage device by using more than one SMS scheme.

CATALOG ALIAS field (CATALOG subsystem parameter)

The value of the CATALOG ALIAS field specifies the high-level qualifier for VSAM data sets that are to be used by the Db2 catalog and directory.

The installation CLIST also uses the value in the CATALOG ALIAS field to customize the following statements in Db2 installation and migration jobs:

- The NAME parameter of the DEFINE ALIAS statement for the optional VSAM ICF user catalog for Db2 data sets (job DSNTIJCA)
- The CATALOG parameter of AMS DELETE and DEFINE statements for Db2 catalog (DSNDB06), Db2 directory (DSNDB01), active logs, and bootstrap data sets (jobs DSNTIJDE, DSNTIJEN, and DSNTIJIN)
- The VCAT parameter of various CREATE STOGROUP statements (jobs DSNTIJTM, DSNTIJSG, DSNTEJ1, DSNTEJ2A, and DSNTEJ3M)

Acceptable values: 1 - 8 characters
Default: DSNCAT
Update: see Changing the high-level qualifier for Db2 data sets
DSNZPxxx: DSN6SPRM CATALOG

VSAM data set cataloging options: The installation jobs classify the Db2 VSAM data sets, which include recovery log, subsystem, and user data sets. You must create the catalog that defines these data sets through the VSAM ICF.

Recommendation: Use an ICF user catalog to classify all Db2 objects because you can use aliases for user catalogs. Multi-level ICF catalog aliases are acceptable. However, when you use the CREATE STOGROUP statement, you might need to use an alias for the VCAT option, which must be a single-level one-to
You can use a master catalog, but only if the name of the master catalog is a single-level name that is one to eight characters.

Ensure that your alias conforms to your local naming conventions. If you are migrating, you should use the same ICF catalog aliases for Db2 12 that you used for your previous version because Db2 12 uses many of the data sets from your previous version that are already classified.

Whether you are installing or migrating, Db2 does not require you to catalog all Db2 VSAM data sets in the same ICF catalog nor use the same prefix or catalog alias. However, data sets for the Db2 catalog (DSNDB06) and Db2 directory (DSNDB01) must have the high-level qualifier that you specify in the CATALOG ALIAS field. See the following table for guidelines for the high-level qualifier of data sets. The BSDS is VSAM KSDS. The archive logs are sequential. All other data sets are VSAM linear data sets.

<table>
<thead>
<tr>
<th>Table 10. Db2 data sets ICF catalog guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Db2 data sets</strong></td>
</tr>
<tr>
<td>Db2 directory (DSNDB01)</td>
</tr>
<tr>
<td>Db2 catalog (DSNDB06)</td>
</tr>
<tr>
<td>Active logs</td>
</tr>
<tr>
<td>Bootstrap data set</td>
</tr>
<tr>
<td>Default database (DSNDB04)</td>
</tr>
<tr>
<td>Work file database</td>
</tr>
<tr>
<td>Archive logs</td>
</tr>
<tr>
<td>User table spaces</td>
</tr>
<tr>
<td>User index spaces</td>
</tr>
</tbody>
</table>

You must provide any catalog connections for log and bootstrap data sets that you catalog apart from the Db2 directory and catalog data sets.

**Recommendation:** Add an alias for the proper catalog.

Although you can catalog two Db2 subsystems in the same ICF catalog, they must not share the same ICF catalog alias because the alias is the only parameter that makes the data set names unique.

**Data set naming conventions:** The value that you specify as the ICF catalog alias is also used as the high-level qualifier for Db2 directory and catalog VSAM data sets.
The data sets for the Db2 directory and catalog databases and the default database are all VSAM linear data sets (LDSs). Their data set names have the following format:

```
dddddd.DSNDbn.bbbbbb.xxxxxxxxx.y0001.Accc
```

With the following definitions:

- `ddddddd`: The high-level qualifier, which is the value that you supply for the CATALOG ALIAS field.
- `DSNDbn`: A constant identifying this as a Db2 data set. For `n`, use C for a cluster name or D for a data component name.
- `bbbbbbb`: The database name. The system database names are:
  - `DSNDB01`: The Db2 directory database
  - `DSNDB04`: The default database
  - `DSNDB06`: The Db2 catalog database
  - `DSNDB07`: The default work file database
- `xxxxxxxx`: The individual table space or index space.
- `y0001.Accc`: The data set. For table spaces and index spaces that can be reorganized with SHRLEVEL CHANGE or SHRLEVEL REFERENCE, `y` can be I or J, depending on whether REORG has been run. For other table spaces and index spaces, `y` is I. `ccc` is the partition number of a partitioned table space or index space, or the relative data set number of a simple or segmented table space or index space.

For example, if the catalog alias is DSNCAT, one of the Db2 directory data sets is named:

```
DSNCAT.DSNDB01.DSNDB01.DB001.Y0001.A001
```

Similarly, one of the Db2 catalog data sets is named:

```
DSNCAT.DSNDB06.DSNDB06.SYSTSVEW.Y0001.A001
```

**DEFINE CATALOG field**

The value of the DEFINE CATALOG field determines whether Db2 is to create a new ICF catalog.

- **Acceptable values:** YES, NO
- **Default:** YES
- **Update:** see [Changing the high-level qualifier for Db2 data sets](#)
- **DSNZPxxx:** none

- **YES**
  
  Builds a new ICF catalog by using the alias that is specified in the CATALOG ALIAS field.
NO  Signals that the catalog that is named in the CATALOG ALIAS field already exists; the CLIST does not create a new one.

If you specify YES, Db2 edits job DSNTIJCA which, when run, creates a user catalog and an alias for that catalog. Db2 creates the high-level qualifier of the catalog name by adding a number to the end of the alias that is defined in the CATALOG ALIAS field. If the alias has fewer than eight characters, Db2 appends a “1” to the end of the alias. For example, if you accept the default of DSNCAT for the CALTAGALIAS field, the catalog that Db2 creates is named DSNCAT1.USER.CATALOG. If the alias has eight characters, Db2 changes the last character into a 1. If the last character is already a 1, Db2 changes the 1 to a 2.

DIRECTORY AND CATALOG DATA field (CATDDACL, CATDMGCL, and CATDSTCL subsystem parameters)

The DIRECTORY AND CATALOG DATA field specifies the explicit Storage Management Subsystem (SMS) classes that are to be used for defining VSAM data sets for the Db2 catalog and directory. Your SMS storage administrator defines these SMS classes. To use ACS routines for defining these data sets, leave this field blank.

For migration, values that you enter in this field initially apply only for defining new Db2 catalog and directory data sets. Existing Db2 data sets are not changed until they are redefined as part of a REORG invoked from the DSNTIJEN job.

<table>
<thead>
<tr>
<th>Column</th>
<th>DATA CLASS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptable values:</td>
<td>Blank or a valid SMS data class name</td>
<td>Blank or a valid SMS management class name</td>
<td>Blank or a valid SMS storage class name</td>
</tr>
<tr>
<td>Default:</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>Update:</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6SPRM CATDDACL</td>
<td>DSN6SPRM CATDMGCL</td>
<td>DSN6SPRM CATDSTCL</td>
</tr>
</tbody>
</table>

A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.

**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

DIRECTORY AND CATALOG INDEXES field (CATXDACL, CATXMGCL, and CATXSTCL subsystem parameters)

The DIRECTORY AND CATALOG INDEXES field specifies the explicit Storage Management Subsystem (SMS) classes that are to be used for defining VSAM data sets for Db2 catalog and directory indexes. Your SMS storage administrator defines these SMS classes. To use ACS routines for defining these data sets, leave this field blank.
For migration, values that you enter in this field initially apply only for defining new data sets for Db2 catalog and directory indexes. Existing Db2 catalog and directory index data sets are not changed until they are redefined as part of a REORG invoked from the DSNTIJEN job.

Table 12. Acceptable values and defaults for columns of DIRECTORY AND CATALOG INDEXES field

<table>
<thead>
<tr>
<th>Column</th>
<th>DATA CLASS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptable values:</td>
<td>Blank or a valid SMS data class name</td>
<td>Blank or a valid SMS management class name</td>
<td>Blank or a valid SMS storage class name</td>
</tr>
<tr>
<td>Default:</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>Update:</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6SPRM CATXDACL</td>
<td>DSN6SPRM CATXMGCL</td>
<td>DSN6SPRM CATXSTCL</td>
</tr>
</tbody>
</table>

A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.

**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

**COMPRESS SPT01 field (COMPRESS_SPT01 subsystem parameter)**

The value of the COMPRESS_SPT01 subsystem parameter specifies whether the SPT01 directory table space is to be compressed.

- **Acceptable values:** NO, YES
- **Default:** NO
- **Update:** option 1 on panel DSNTIPB
- **DSNZPxxx:** DSN6SPRM COMPRESS_SPT01

**NO** The SPT01 table space is not compressed. This is the default value.

**YES**

The SPT01 table space is compressed during the next reorganization of the table space.

In a data sharing environment, all members should use the same setting for the COMPRESS_SPT01 parameter. The result of a table space reorganization will be determined by the value of the COMPRESS_SPT01 parameter on the member that executes the REORG TABLESPACE utility.

**COMPRESS Db2 DIR LOBS field (COMPRESS_DIRLOB subsystem parameter)**

The value of the COMPRESS_DIRLOB subsystem parameter specifies whether LOB table spaces in the Db2 directory are to be compressed.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.
Acceptable values: NO, YES
Default: NO
Data sharing scope: All members should use the same setting
Online changeable: Yes
Update: option 1 on panel DSNTIPB
DSNZPxxx: DSN6SPRM COMPRESS_DIRLOB

**NO**  LOB table spaces in the Db2 directory are not compressed. This is the default value.

**YES**  LOB table spaces in the Db2 directory are compressed the next time they are reorganized.

In a data sharing environment, all members should use the same setting for the COMPRESS_DIRLOB parameter. The result of a table space reorganization will be determined by the value of the COMPRESS_DIRLOB parameter on the member that executes the REORG TABLESPACE utility.

**Note:** Compressing the LOB table spaces in the Db2 directory requires zEnterprise® Data Compression hardware. The z/OS environment needs to have the zEDC card installed in order for LOB compression to occur.

**SPT01 INLINE LENGTH field (SPT01_INLINE_LENGTH subsystem parameter)**

The SPT01_INLINE_LENGTH subsystem parameter specifies the maximum length of LOB column data in the SPT01 directory table space that is to be maintained in the base table. The length is specified in single-byte characters.

In a data sharing environment, all members must use the same setting for this parameter.

Acceptable values: NOINLINE, 1 - 32138
Default: 32138
Update: option 1 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SPT01_INLINE_LENGTH

To optimize performance, use the largest possible setting for this parameter, which will decrease the likely number of fetches from the SPT01 LOB auxiliary tables. A large value also increases the efficiency of SPT01 compression when you specify YES for the COMPRESS_SPT01 subsystem parameter. Use a smaller value for the SPT01_INLINE_LENGTH parameter to increase the capacity for storing additional rows in SPT01 if the size of SPT01 exceeds the maximum of 64 GB even with compression activated.

A change to this parameter does not take effect until you issue the SET SYSPARM command. In other words, even if you start or restart Db2 after changing the value, the change is not honored until you issue the SET SYSPARM command. If you increase the setting, Db2 places SPT01 in REORG-pending advisory status (AREO*) after you issue the SET SYSPARM command. If you decrease the setting, Db2 places SPT01 in REORG-pending restrictive status (REORP) after you issue the SET SYSPARM command.
If you change the setting of SPT01_INLINE_LENGTH for a data sharing member, change the setting on all members before running the -SET SYSPARM command on any member.

**DSNTIPA3: Data parameters panel**

The entries on the DSNTIPA3 panel define the storage management preferences for the storage groups and data sets that are created during installation and migration.

For information about updating parameters with changes that cannot be made by using the panels, see [Updating subsystem parameter and application default values](#).

---

**PERMANENT UNIT NAME field**

The PERMANENT UNIT NAME field specifies the device type or unit name that is to be used for allocating data sets that are created during installation or migration.

Acceptable values: blank, valid device type, unit name

Default: 3390

Update: none

DSNZPxxx: none

The device type or unit name applies to the following data sets:
- ICF catalog
- `prefix.DBRMLIB.DATA.suffix`
- `prefix.RUNLIB.LOAD.suffix`
- `prefix.SRCLIB.DATA.suffix`
- The two data sets that the DSNTINST CLIST generates:
  - `prefix.NEW.SDNSNTEMP`
  - `prefix.NEW.SDNSAMP`
The value of PERMANENT UNIT NAME identifies a direct access unit name for partitioned data sets and the ICF catalog. If you want to use different device types for different data sets, edit the installation or migration jobs after you complete the tailoring session. A common device type is 3390.

The value of PERMANENT UNIT NAME is sometimes used during IVP processing to place output (from COPY TABLESPACE, for example) on the device type that is specified for the PERMANENT UNIT NAME field.

A change to PERMANENT UNIT NAME during migration does not affect the ICF catalog, Db2 catalog, directory, or logs. The new value is used for data sets that are created during migration.

**TEMPORARY UNIT NAME field**

The value of the TEMPORARY UNIT NAME field specifies the device type or unit name that is to be used for allocating temporary data sets.

- **Acceptable values:** blank, or valid device type or unit name
- **Default:** SYSDA
- **Update:** none
- **DSNZPxxx:** none

The value of TEMPORARY UNIT NAME is the direct access or disk unit name that is used for the precompiler, compiler, assembler, sort, linkage editor, and utility work files in the tailored jobs and CLISTs.

Be aware that this field was previously used to specify the VOLTDEVT subsystem parameter, which provides the device type that is used by Db2 utilities that dynamically allocate temporary data sets. You now specify the VOLTDEVT parameter in the TEMPORARY DS UNIT NAME field on the DSNTIP61: Db2 utilities parameters panel 2.

**DB2 zHyperLinks SCOPE field (ZHYPERLINK subsystem parameter)**

The ZHYPERLINK subsystem parameter specifies the default scope of the zHyperLink protocol for I/O requests.

- **Acceptable values:** DISABLE, ENABLE, DATABASE or ACTIVELOG
- **Default:** DISABLE
- **Update:** Option 3 on panel DSNTIPA3
- **DSNZPxxx:** ZHYPERLINK

The following values can be used to specify the default value for the ZHYPERLINK parameter:

- **DISABLE**
  - Db2 does not use zHyperLink for any I/O requests.

- **ENABLE**
  - All eligible I/O requests that Db2 makes request the zHyperLink protocol.

- **DATABASE**
  - Db2 requests the zHyperLink protocol only for database synchronous read I/Os.

- **ACTIVELOG**
  - Db2 requests the zHyperLink protocol only for active log write I/Os.
Note: In order for the zHyperLink protocol to be enabled for I/O requests by the Db2 system, in addition to specifying a value of ENABLE, DATABASE, or ACTIVELOG for the ZHYPERLINK parameter, the SMS storage class of the Db2 data sets must be enabled for zHyperLink eligibility using the Integrated Storage Management Facility (ISMF).

Related concepts:
- Application compatibility levels in Db2 (Db2 Application programming and SQL)

Related tasks:
- "Migration step 29: Activate function level 500 or higher" on page 538

Related reference:
- APPLCOMPAT bind option (Db2 Commands)

**CLIST ALLOCATION field**
The CLIST ALLOCATION field controls the storage management properties for the prefix.NEW.SDSNTEMP and prefix.NEW.SDNSAMP data sets, which the CLIST creates during the tailoring session.

<table>
<thead>
<tr>
<th>Table 13. Acceptable values and defaults for field columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
</tr>
<tr>
<td>Acceptable values:</td>
</tr>
<tr>
<td>Default:</td>
</tr>
<tr>
<td>Update:</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
</tr>
</tbody>
</table>

A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.

**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

**NON-VSAM DATA field**
The values of the NON-VSAM DATA field specify the storage management properties for non-VSAM data. Specifically, the NON-VSAM DATA field applies to the prefix.DBRMLIB.DATA, prefix.RUNLIB.LOAD, and prefix.SRCLIB.DATA data sets.

<table>
<thead>
<tr>
<th>Table 14. Acceptable values and defaults for field columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
</tr>
<tr>
<td>Acceptable values:</td>
</tr>
<tr>
<td>Default:</td>
</tr>
</tbody>
</table>
A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.

**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

### VSAM CATALOG, DEFAULT AND WORK FILE DATABASE field

The values in the VSAM CATALOG, DEFAULT AND WORK FILE DATABASE field specify storage management properties. These properties apply to temporary data sets, the default storage group and sample storage group, and the VSAM catalog (if a new one is created).

---

**Table 14. Acceptable values and defaults for field columns (continued)**

<table>
<thead>
<tr>
<th>Column</th>
<th>VOL/SER</th>
<th>DATA CLASS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update:</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

---

**Table 15. Acceptable values and defaults for field columns**

<table>
<thead>
<tr>
<th>Column</th>
<th>VOL/SER</th>
<th>DATA CLASS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptable values:</td>
<td>Blank or a valid z/OS volume serial number</td>
<td>Blank or a valid SMS data class name</td>
<td>Blank or a valid SMS management class name</td>
<td>Blank or a valid SMS storage class name</td>
</tr>
<tr>
<td>Default:</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td>Update:</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

The CREATE STOGROUP statements for the default storage group and sample storage groups are tailored to include or exclude the VOLUMES, DATACLAS, MGMTCLASS, and STORCLASS parameters according to whether you specify arguments for them in this field.

- If a column is not blank, the corresponding parameter is included, with the value as its argument.
- If a column is blank, the corresponding parameter is excluded.
- If all columns are blank, the VOLUMES parameter is included, with the asterisk as its argument.

---

A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.
**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

For migration, this field has no effect on the VSAM catalog or the default storage group because these objects are provided by the previous release of Db2.

**DSNTIPA4: BSDS and active log panel**

The entries on the DSNTIPA4 panel define the name of the ICF catalog for the Db2 bootstrap data set (BSDS) and active log VSAM datasets. The entries also define optional z/OS Storage Management Subsystem (SMS) classes and disc volume serial names that are to be used when defining these data sets.

---

### BSDS AND ACTIVE LOG ICF CATALOG ALIAS field

Indicates the dedicated ICF catalog alias for Db2 bootstrap data sets (BSDS) and active log VSAM data sets.

<table>
<thead>
<tr>
<th>BSDS AND ACTIVE LOG ICF CATALOG ALIAS</th>
<th>VOL/SER</th>
<th>DATA CLASS</th>
<th>SMS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 BSDS AND ACTIVE LOG ICF CATALOG ALIAS</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
</tr>
<tr>
<td>2 LOG COPY 1, BSDS 2</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
</tr>
<tr>
<td>3 LOG COPY 2, BSDS 1</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
<td>==&gt;</td>
</tr>
</tbody>
</table>

**Acceptable values:** 1 - 8 alphanumeric characters, beginning with a letter or alphabet extender

**Default:** See Table 16 on page 89

**Update:** option 3 on panel DSNTIPB

**DSNZPxxx:** none

The initial value of the BSDS AND ACTIVE LOG ICF CATALOG ALIAS field and how its value sets the high level qualifier of the BSDS, active log, and archive log data set names on panel DSNTIPH depend on the use case:
Table 16. Use cases for the BSDS AND ACTIVE LOG ICF CATALOG ALIAS field

<table>
<thead>
<tr>
<th>INSTALL</th>
<th>TYPE on panel</th>
<th>Data sharing?</th>
<th>Can be changed?</th>
<th>Initial value source</th>
<th>This field used as high-level qualifier for BSDS, active log, and archive log data set names on DSNTIPH?</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTALL No</td>
<td>Yes</td>
<td>DSNTIDXn VBSDSLOG parameter, if it is not NULL; otherwise, the CATALOG ALIAS field on panel DSNTIPA2 (CATALOG AND DIRECTORY)</td>
<td>Yes, if this field is different than the CATALOG ALIAS field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTALL First member</td>
<td>Yes</td>
<td>DSNTIDXn VBSDSLOG parameter, if it is not NULL; otherwise, the CATALOG ALIAS field on panel DSNTIPA2 (CATALOG AND DIRECTORY)</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTALL Subsequent member</td>
<td>No</td>
<td>DSNTIDXn VBSDSLOG parameter</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIGRATE Any</td>
<td>No</td>
<td>CATALOG ALIAS on panel DSNTIPA2</td>
<td>No, input from the DSNTIDXn member is used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPDATE N/A</td>
<td>No</td>
<td>DSNTIDXn VBSDSLOG parameter</td>
<td>No, input from the DSNTIDXn member is used</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related reference:
[DSNTIPH: System resource data set names panel]

LOG COPY 1, BSDS 2 field
When installing Db2, you can change the values of the LOG COPY 1, BSDS 2 field. Specify the storage management properties for the first copy of the active log data set and the second copy of the bootstrap data set (BSDS).

When migrating Db2, you cannot change any values in this field.

This field has four columns where you can indicate the volume serial number and the SMS classes for data, management, and storage that are to be used for defining particular data sets or storage groups. The acceptable values and default values for each column are shown in the following table. All fields are optional.

Edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

Also edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

Table 17. Acceptable values and defaults for field columns

<table>
<thead>
<tr>
<th>Column</th>
<th>VOL/SER</th>
<th>DATA CLASS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptable values:</td>
<td>Blank or a valid z/OS volume serial number</td>
<td>Blank or a valid SMS data class name</td>
<td>Blank or a valid SMS management class name</td>
<td>Blank or a valid SMS storage class name</td>
</tr>
<tr>
<td>Default:</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
</tbody>
</table>
A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.

**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

**LOG COPY 2, BSDS 1 field**

When installing Db2, you can change the values of the LOG COPY 2, BSDS 1 field. Use this field to specify the storage management properties for the second copy of the active log and the first copy of the BSDS.

When migrating Db2, you cannot change any values in this field.

This field has four columns where you can indicate the volume serial number and the SMS classes for data, management, and storage that are to be used for defining particular data sets or storage groups. The acceptable values and default values for each column are shown in the following table. All fields are optional.

Edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

Also edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

### Table 18. Acceptable values and defaults for field columns

<table>
<thead>
<tr>
<th>Column</th>
<th>VOL/SER</th>
<th>DATA CLASS</th>
<th>MGMT CLASS</th>
<th>STOR CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Acceptable values:</strong></td>
<td>Blank or a valid z/OS volume serial number</td>
<td>Blank or a valid SMS data class name</td>
<td>Blank or a valid SMS management class name</td>
<td>Blank or a valid SMS storage class name</td>
</tr>
<tr>
<td><strong>Default:</strong></td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
<td>Blank</td>
</tr>
<tr>
<td><strong>Update:</strong></td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td><strong>DSNZPxxx:</strong></td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

A blank value in a column directs the CLIST to remove the corresponding storage-management-related parameters from all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field. A non-blank value is used by the CLIST to customize the corresponding storage-management-related parameters in all JCL, AMS, ALLOC, and DDL statements that are used to create data sets and storage groups that are associated with the field.
**Recommendation:** To help performance and recovery, avoid dependency on a single storage device by using an SMS scheme that uses more than one volume serial number.

**DSNTIP00: Activate a function level panel**

Use the DSNTIP00 panel to specify a function level to activate and related settings.

```
DSNTIP00 ACTIVATE A FUNCTION LEVEL FOR DB2

Enter the current and target DB2 function levels:
1 CURRENT FUNCTION LEVEL ==> (DB2's current function level)
2 TARGET FUNCTION LEVEL ==> V12R1M500 (Function level to be activated)

Enter the default application compatibility level:
3 APPL COMPAT LEVEL ==> V12R1M500 (DB2 function level or V11R1 or V10R1)

Enter the default SQL level for the precompiler:
4 PRECOMPILE SQL LEVEL ==> V12R1M500 (DB2 function level or V11R1 or V10R1)

PRESS: ENTER to continue RETURN to exit HELP for more information
```

The purpose of this panel depends on the value of the USE Z/OSMF WORKFLOW field on panel DSNTIP1A:

- When the USE Z/OSMF WORKFLOW field value is NO, the CLIST uses the settings specified on this panel to customize jobs for migration to Db2 12 at function level V12R1M100. Then you can use the ACTIVATE function of the CLIST to activate function level 500 or higher.
- When the USE Z/OSMF WORKFLOW field value is YES, the CLIST uses the settings specified on this panel to customize jobs for activating the specified Db2 function level and for setting the specified APPLCOMPAT and SQLLEVEL settings after completion of migration to Db2 12 function level. For z/OSMF migrations, all fields on this panel are selected as z/OSMF variables and the settings are stored in DSNTIWMN, the variable input file for the DSNTIWMN workflow. You can use the DSNTIWMN to complete the migration of a subsystem or data sharing group to function level 500 or higher. In Db2 data sharing, do not run DSNTIWMN before all members of the group are migrated to Db2 12 at function level V12R1M100.

How you use CURRENT FUNCTION LEVEL and TARGET FUNCTION LEVEL fields of this panel depend on your activity. In some situations the fields are protected and cannot be changed. The following table describes how the values are used in particular situations.

<table>
<thead>
<tr>
<th>Activity</th>
<th>CURRENT FUNCTION LEVEL field</th>
<th>TARGET FUNCTION LEVEL field (FUNCLVL)</th>
<th>JCL templates updated (result value)</th>
</tr>
</thead>
</table>
| Installing a new Db2 subsystem or originating data sharing member by using z/OSMF or not. | The field is blank and protected. | The default is V12R1M500. You can modify the value to specify a higher function level. | • DSNTIJTC (FUNCLVL)  
• DSNTIJAF (FUNCLVL) |
<p>| Installing subsequent data sharing members, including by using z/OSMF | The field is blank and protected. | The default is FUNCLVL from the DSNTIIDxx output member from the first group member. | None. |</p>
<table>
<thead>
<tr>
<th>Activity</th>
<th>CURRENT FUNCTION LEVEL field</th>
<th>TARGET FUNCTION LEVEL field (FUNCLVL)</th>
<th>JCL templates updated (result value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabling data sharing</td>
<td>The field is blank and protected.</td>
<td>The default is FUNCLVL from the DSNTIDxx member from the first group member.</td>
<td>None.</td>
</tr>
<tr>
<td>Migrating a Db2 subsystem or the first data sharing member to Db2 12</td>
<td>The field is blank and protected.</td>
<td>The default value is V12R1M100 and it is protected.</td>
<td>• DSNTIJTC (catalog level V12R1M500)</td>
</tr>
<tr>
<td>Migrating subsequent group members to Db2 12</td>
<td>The field is blank and protected.</td>
<td>The default value is V12R1M100 and it is protected.</td>
<td>None.</td>
</tr>
<tr>
<td>Activating function levels</td>
<td>The default is FUNCLVL from DSNTIDxx</td>
<td>The default is blank. You can specify V12R1M500 or higher.</td>
<td>• DSNTIJJA1 (V12R1M501)</td>
</tr>
<tr>
<td>Using z/OSMF to migrate a Db2 subsystem or the first data sharing member to Db2 12</td>
<td>The field is blank and protected.</td>
<td>The default is V12R1M500. You can modify the value to specify higher function levels.</td>
<td>• DSNTIJJA1 (V12R1M501)</td>
</tr>
<tr>
<td>Using z/OSMF to migrate subsequent group members to Db2 12</td>
<td>The field is blank and protected.</td>
<td>The default is FUNCLVL from the DSNTIDxx output member from the first group member, and the field is protected.</td>
<td>None.</td>
</tr>
<tr>
<td>Using z/OSMF to activate a function level</td>
<td>The default is FUNCLVL from the input DSNTIDxx member. You can specify V12R1M100 or higher.</td>
<td>The field is blank. You can specify V12R1M100 or higher.</td>
<td>• DSNTIJTC (FUNCLVL)</td>
</tr>
<tr>
<td>Updating subsystem parameters</td>
<td>The field is blank and protected.</td>
<td>The default is FUNCLVL from DSNTIDxx, and the field is protected.</td>
<td>None.</td>
</tr>
</tbody>
</table>

Notes:
1. DSNTIJJA1 is generated only if CURRENT FUNCTION LEVEL is V12R1M100 and TARGET FUNCTION LEVEL is V12R1M502 or higher.
2. DSNTIJJA1 is generated only if TARGET FUNCTION LEVEL is V12R1M502 or higher.

Related tasks:
- [Activating Db2 12 function levels (Db2 for z/OS What’s New?)](#)

Related reference:
- USE Z/OSMF WORKFLOW field
- DSNTIPA1 or DSNTILA1: Main panel
- INSTALL TYPE field

Related information:
- PI70406: Add ACTIVATE mode to the Db2 installation CLIST

**CURRENT FUNCTION LEVEL field**
The value of the CURRENT FUNCTION LEVEL field indicates the currently activated Db2 function level for certain situations.
Acceptable values: function-level
Default: Blank or the FUNCLVL value from the input DSNTIDxx member.
Update None
DSNZPxxx: None

**function-level**

The current activated function level. The format is VvvRrMmmm, where vvv is the version, r is the release, and mmmm is the modification level. For example, V12R1M504 identifies function level 504. For a list of all available function levels in Db2 12, see [What's new in Db2 12 function levels (Db2 for z/OS What's New?)](#).

How the CLIST uses this field depends on the INSTALL TYPE value that you specify on panel DSNTIP1:

- For ACTIVATE, the CURRENT FUNCTION LEVEL field is set to the FUNCLVL value from the input DSNTIDxx member. You can change the value. The CLIST uses this field and the TARGET FUNCTION LEVEL field to generate the correct set of jobs required to activate the specified target function level.
- For other INSTALL TYPE values, the CURRENT FUNCTION LEVEL field is blank and cannot be changed.

**z/OSMF migration**

The CURRENT FUNCTION LEVEL field value is never a z/OSMF variable and cannot be selected as such.

Related concepts:

- [Function levels and related levels in Db2 12 (Db2 for z/OS What's New?)](#)

Related tasks:

- [Activating Db2 12 function levels (Db2 for z/OS What's New?)](#)
- [Automating Db2 migration by using z/OS Management Facility](#)

Related reference:

- [ACTIVATE (Db2) (Db2 Commands)](#)

**TARGET FUNCTION LEVEL field**

The value of the TARGET FUNCTION LEVEL field specifies a Db2 function level to activate.

Acceptable values: function-level
Default: Situation dependent. See [DSNTIP00: Activate a function level panel](#)
Update None
DSNZPxxx: None

**function-level**

The value used in the -ACTIVATE command in the DSNTIJAF job. The format is VvvRrMmmm, where vvv is the version, r is the release, and mmmm is the modification level. For example, V12R1M504 identifies function level 504. For a list of all available function levels in Db2 12, see [What's new in Db2 12 function levels (Db2 for z/OS What's New?)](#)

Changing function levels does not always change the catalog level but you should always run the DSNTIJTC job before running the DSNTIJAF job. If the
current catalog level supports the requested function level, DSNTIJTC ends
with return code 4 and informational messages. Activating a lower function
level never changes the catalog level.

How the CLIST uses this field depends on the INSTALL TYPE value that you
specify on panel DSNTIPA1:

- For MIGRATE when z/OSMF is used, this field is always selected as a z/OSMF
  variable, and the value specifies the function level that is activated at the end of
  the z/OSMF migration process.
- For MIGRATE when z/OSMF is not used, the CLIST completes migration to
  Db2 12 at function level V12R1M100.
- For INSTALL or ACTIVATE, the CLIST applies the value of this field to the
  -ACTIVATE FUNCTION LEVEL command in the DSNTIJAF job and in the
  CATMAINT utility control statement in the DSNTIJTC job.
- For UPDATE, the CLIST ignores this field.

z/OSMF migration

The DB2 FUNCTION LEVEL field value is always selected as a z/OSMF variable
and the setting indicate the target Db2 12 level after the migration process
completes.

Related concepts:
- Function levels and related levels in Db2 12 (Db2 for z/OS What's New?)

Related tasks:
- Activating Db2 12 function levels (Db2 for z/OS What's New?)
- Automating Db2 migration by using z/OS Management Facility

Related reference:
- -ACTIVATE (Db2) (Db2 Commands)

APPL COMPAT LEVEL field (APPLCOMPAT subsystem
parameter)

The APPL COMPAT LEVEL field specifies the default value of the APPLCOMPAT
bind option for BIND and REBIND commands.

This subsystem parameter is the default value of the APPLCOMPAT bind option
for any BIND or REBIND command that does not explicitly specify that option.

Tip: Do not raise the default application compatibility level of the Db2 subsystem
immediately after migrating or activating a new function level. Instead, wait until
applications have been verified to work correctly at the higher function level, and
any incompatibilities have been resolved. For details, see Enabling default
application compatibility with function level 500 (Db2 Application programming
and SQL).

Tip: Set all Db2 data sharing members to the same value.

Acceptable values: function-level, V11R1, V10R1
Default: V11R1 or V10R1, as specified in the CLIST migration input
member of the Db2 subsystem or first member of the data
sharing group; V12R1M500 for installation
Update: option 5 on panel DSNTIPB
DSNZIPxxx: DSN6SPRM.APPLCOMPAT
The following values can be used to specify the default value for the
APPLCOMPAT bind option for packages:

function-level

Specifies the application compatibility behavior of the function-level function
level. The equivalent function level or higher must be activated.

The format is VvvRrmmm, where v is the version, r is the release, and mmm is
the modification level. For example:

V12R1M504

Identifies compatibility with the current highest available function
level, function level 504.

V12R1M500

Identifies compatibility with the function level that enables new
function in the initial Db2 12 release, function level 500. V12R1M500 is
the same as V12R1.

V12R1M100

Identifies compatibility with the function level before you activate new
function, after migration to Db2 12, function level 100. V12R1M100 is
the same as V11R1.

For a list of supported function-level values, see What's new in Db2 12 function
levels (Db2 for z/OS What's New?).

V11R1

Specifies Db2 11 compatibility behavior.

V10R1

Specifies Db2 10 compatibility behavior.

z/OSMF migration

The APPL COMPAT LEVEL field value is always selected as a z/OSMF variable
and the setting indicate the target application compatibility level for Db2 after the
migration process completes.

Related concepts:

- Application compatibility levels in Db2 (Db2 Application programming and
  SQL)
- Function levels and related levels in Db2 12 (Db2 for z/OS What's New?)

Related tasks:

- Controlling Db2 application compatibility (Db2 for z/OS What's New?)
- “Migration step 29: Activate function level 500 or higher” on page 538
- Automating Db2 migration by using z/OS Management Facility

Related reference:

- APPLCOMPAT bind option (Db2 Commands)

PRECOMPILIER SQL LEVEL field (SQLLEVEL DECP value)

Specifies the Db2 function level or release to be used by the precompiler and
coprocessor for determining the allowed SQL syntax in the absence of an explicit
specification of the SQLLEVEL option. Db2 also uses this parameter to set the
SYSIBM.DEFAULT_SQLLEVEL global variable. In most cases set this field to
current Db2 function level.
Acceptable values:
V10R1, V11R1, function-level

Default:
Installation: V12R1M500
z/OSMF migration: V12R1M500
Migration: V12R1M100

Update:
option 5 on panel DSNTIPB.

DSNZPxxx:
DSNHDECPSQLLEVEL

function-level
The precompiler and coprocessor accept only SQL syntax that is valid in a Db2 12 function level. The format is VvvRrmmm, where Vv is the version, r is the release, and mmm is the modification level.

V11R1 or V12R1M100
The precompiler and coprocessor accept only SQL syntax that is valid in Db2 11 new-function mode.

V10R1
The precompiler and coprocessor accept only SQL syntax that is valid in Db2 10 new-function mode.

The function level activated on the Db2 subsystem does not restrict the SQLLEVEL value. However, you must ensure that you bind the resulting DBRM with the correct application compatibility level on a Db2 subsystem with the correct function level activated.

An online update to SQLLEVEL DECP parameter has immediate effect for the precompiler and coprocessor. However the SYSIBM.DEFAULT_SQLLEVEL global variable update requires a restart of Db2.

z/OSMF migration
The SQLLEVEL DECP value is always selected as a z/OSMF variable and the setting is used to indicate the target SQLLEVEL level for Db2 after the migration process completes.

Related concepts:
[Function levels and related levels in Db2 12 (Db2 for z/OS What's New?)]

Related tasks:
[Automating Db2 migration by using z/OS Management Facility]

Related reference:
[Built-in global variables (Db2 SQL)]

DSNTIPK: Define group or member panel
The DSNTIPK panel follows panel DSNTIPA2 when you select a data sharing function (GROUP, MEMBER, or ENABLE).

You must start Db2 and IRLM group names with an alphabetic character. You should carefully consider the naming convention for a data sharing system. See [Naming in Db2 data sharing] for guidance on planning a naming convention before you choose names for the fields on panel DSNTIPK.
GROUP NAME field (GRPNAME subsystem parameter)
The GRPNAME subsystem parameter identifies the name of a new or existing Db2 data sharing group that is to be used for this subsystem.

Acceptable values: 1 - 8 characters consisting of letters A - Z, numbers 0 - 9, $, 

Default: DSNCAT

DSNZPxxx: DSN6GRP GRPNAME

The group name encompasses the entire data sharing group and forms the basis for the coupling facility structure names.

To avoid names that IBM uses for its z/OS cross-system coupling facility (XCF) groups, the first character must be an uppercase letter J - Z unless the name begins with DSN. Do not use SYS as the first three characters, and do not use UNDESIG as the group name.

MEMBER NAME field (MEMBNAME subsystem parameter)
The MEMBNAME subsystem parameter specifies the name of a new or existing Db2 data sharing member that is to be used for this subsystem.

Acceptable values: 1 - 8 characters

Default: DSN1

DSNZPxxx: DSN6GRP MEMBNAME

Recommendation: Use the z/OS subsystem name. Db2 uses this name as its XCF member name. An example of a member name is DB1G. The member name can consist of letters A - Z, numbers 0 - 9, $, #, and @.

WORK FILE DB field
The value of the WORK FILE DB field specifies the name of the work file database that is to be used for the Db2 member.

Acceptable values: 1 - 8 characters

Default: DSN1

DSNZPxxx: none
Each Db2 member has its own work file database (called DSNDDB07 in a
non-data-sharing environment). One member of the data sharing group can have
the name DSNDDB07, but the recommendation is that you use a more meaningful
name, such as WRKDSN1. You cannot specify a name that begins with DSND
unless the name is DSNDDB07.

**GROUP ATTACH field (SSID DECP value)**
The value of the GROUP ATTACH field specifies a generic group attachment
name. This name is to be used for batch programs, the call attachment facility
(CAF), the RR5 attachment facility (RRSAF), IMS, CICS Transaction Server for
z/OS, and utilities.

Acceptable values: 1 - 4 characters
Default: none
dsnhdecp: SSID

An example of a group attachment name is DB0G. The value that you specify here
is also used in the IEFSSNxx member of SYS1.PARMLIB.

If you leave this field blank, the SSID DECP value is set to the value that is
specified in the SUBSYSTEM NAME field on panel DSNTIPM.

**Related concepts:**
Group attachment names and subgroup attachment names

**GROUP ATTACH field**
The value of the GROUP ATTACH field specifies a generic group attachment
name. This name is to be used for batch programs, the call attachment facility
(CAF), the RR5 attachment facility (RRSAF), IMS, CICS Transaction Server for
z/OS, and utilities.

Acceptable values: 1 - 4 characters
Default: none
dsnhdecp: SSID

An example of a group attachment name is DB0G. The value that you specify here
is also used in the IEFSSNxx member of SYS1.PARMLIB.

If you leave this field blank, the SSID DECP value is set to the value that is
specified in the SUBSYSTEM NAME field on panel DSNTIPM.

**SUBSYS ATTACH field**
The SUBSYS ATTACH field identifies the name of a new or existing subgroup
attachment that is to be associated with this subsystem. A subgroup attachment
name can be specified with a group attachment name to organize how TSO, CAF,
RRSAF, JDBC, ODBC, and Db2 utilities connections find an active Db2 subsystem.

Acceptable values: blank, 1 - 4 characters. Characters must be letters A – Z,
numbers 0 – 9, $, #, or @
Default: blank
Update: modify IEFSSNxx and IPL
DSNZPxxx: none

The value that you specify for the SUBSYS ATTACH field is used in the IEFSSNxx
member of SYS1.PARMLIB.

The default value, blank, means that the member that is specified in the MEMBER
NAME field is not to be associated with a subgroup attachment. If a subgroup
attachment name is entered in the SUBSYS ATTACH field, a group attachment
name must also be specified in the GROUP ATTACH field. A subgroup attachment
can belong to only one group attachment and cannot have the same name as the
group attachment. A member can belong to one subgroup attachment at most.

**Related concepts:**
Group attachment names and subgroup attachment names

Related tasks:

Registering the command prefixes, member group attachment name, and subgroup attachment name

RANDOM ATTACH field (RANDOMATT subsystem parameter)

The RANDOMATT subsystem parameter specifies whether this Db2 member can be used for randomized group attachment or subgroup attachment processing. This setting is ignored when the GROUP ATTACH field is blank.

Acceptable values: YES, NO
Default: YES
DSNZPxxx: DSN6GRP RANDOMATT

YES

Allows this member to be used, at random, for group attachments or subgroup attachments.

NO

Prevents this member from being considered for group attachments or subgroup attachments unless no other members of the defined group are available.

Related concepts:

Group attachment names and subgroup attachment names

DEL CF STRUCTS field (DEL_CFSTRUCTS_ON_RESTART subsystem parameter)

The DEL_CFSTRUCTS_ON_RESTART parameter specifies whether, during restart, Db2 is to attempt to delete the SCA, IRLM lock structure, and any allocated group buffer pools from the coupling facility.

Acceptable values: NO, YES
Default: NO
Update: option 4 on DSNTIPB
DSNZPxxx: DSN6SYSP:DEL_CFSTRUCTS_ON_RESTART

NO

During restart, Db2 does not attempt to delete the data sharing group’s coupling facility structures.

Recommendation: Use this setting for production systems.

YES

During restart, Db2 attempts to delete the data sharing group’s coupling facility structures. The deletion is successful only if no other Db2 subsystems are connected to the structures when the deletion is attempted. If the deletion is successful, the deleted structures are recovered as necessary during a group restart or group buffer pool RECOVER-pending (GRECP) recovery process. This process might increase the amount of time that Db2 takes to complete a restart.

Recommendation: Use this setting for disaster recovery systems.
Use the same setting for the DEL CF STRUCTS field on all members of a data sharing group.

**PEER RECOVERY field (PEER_RECOVERY subsystem parameter)**
The PEER_RECOVERY subsystem parameter specifies whether the data sharing member participates in data sharing peer recovery.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: NONE, RECOVER, ASSIST, BOTH
Default: NONE
DSNZPxxx: DSN6GRP:PEER_RECOVERY

The PEER_RECOVERY subsystem parameter accepts the following values:

- **NONE**
  This member does not participate in peer recovery. Use this option if you have configured the z/OS Automatic Recovery Manager (ARM) to restart failed Db2 members. NONE is the default value.

- **RECOVER**
  This member is recovered by a peer member if it fails.

- **ASSIST**
  This member attempts to initiate peer recovery for other failed members. When this member detects a failure, it attempts to initiate a LIGHT(YES) restart for the failed member if it has not already been initiated, to recover the retained locks.

- **BOTH**
  Both RECOVER and ASSIST options are activated for this member.

Related concepts:

- [Peer recovery for data sharing members (Db2 Data Sharing Planning and Administration)]

**DSNTPG: Installation preferences panel**
The entries on the DSNTPG panel specify the authorization IDs for installation and IVP jobs.
The ROUTINES CREAT OR field specifies the CURRENT SQLID setting that is to be used when creating, configuring, and validating most Db2 supplied routines. This field also specifies the default OWNER for binding packages for these routines if the INSTALL PKG OWNER field is blank.

Acceptable values: 1 - 8 characters, the first of which must be an alphabetic character

Default: SYSADM

Update: option 6 on panel DSNTIPB

The value in the ROUTINES CREAT OR field is used by the installation CLIST to set the value of the AUTHID parameter for installation programs DSNTRIN in job DSNTIJRT and DSNTRVFY in job DSNTIJRV. If the INSTALL PKG OWNER field is not specified, the value in the ROUTINES CREAT OR field is also used by the installation CLIST to set the value of the DEFPKOWN parameter for DSNTRIN and the PKOWNER parameter for DSNTRVFY.

DSNTRIN uses the value of the AUTHID parameter as the CURRENT SQLID for creating and configuring Db2 supplied routines. It also uses the value of the DEFPKOWN parameter to determine the default OWNER for binding the package for a Db2 supplied routine when an owner is not explicitly specified through the PKOWNER parameter in the routine's configuration control statement. If the DEFPKOWN parameter is not present, DSNTRIN uses the AUTHID parameter to determine the default OWNER.

DSNTRVFY uses the value of the AUTHID parameter as the CURRENT SQLID for validating Db2 supplied routines. It also uses the value of the PKOWNER parameter as the default OWNER for binding packages for routine validation. If the DEFPKOWN parameter is not present, DSNTRVFY uses the AUTHID parameter to determine the OWNER.
If you use the installation SYSOPR authority for Db2 installation or migration, set the ROUTINES CREAT field to SYSINSTL. Also, set the SEC DEF CREAT and INSTALL SQL ID fields to SYSINSTL and the INSTALL PKG OWNER field to a non-blank value. Db2 marks a routine system-defined when it is created or configured by a user who has the installation SYSADM authority, the installation SYSOPR authority, or both and when the CURRENT SQLID is set to SYSINSTL.

**Related reference:**

- [SYSROUTINES catalog table (Db2 SQL)](#)

### SEC DEF CREAT field

The SEC DEF CREAT field specifies the CURRENT SQLID value for creating and configuring Db2-supplied routines that are defined with the SECURITY DEFINER option.

Acceptable values: 1 - 8 characters, the first of which must be an alphabetic character

Default: SYSADM

Update: option 6 on panel DSNTIPB

DSNZPxxx: none

The value that you enter in the SEC DEF CREAT field is assigned by the installation CLIST as the setting of the SECDEFID parameter for installation program DSNTRIN in job DSNTIJRT.

The SECDEFID parameter is used by the DSNTRIN program as the CURRENT SQLID setting when creating and configuring Db2-supplied routines that have the SECURITY DEFINER attribute. The default OWNER of any packages bound for the routines is specified in the INSTALL PKG OWNER field. If the INSTALL PKG OWNER field is not specified, the default OWNER is the value of the ROUTINES CREAT field.

If you use the installation SYSOPR authority for Db2 installation or migration, set the SEC DEF CREAT field to SYSINSTL. You must also set the ROUTINES CREAT and INSTALL SQL ID fields to SYSINSTL and define SYSINSTL in RACF.

**Requirement:** Specify an ID that can be used as a logon ID because it will be used by WLM to execute routines that have the SECURITY DEFINER attribute.

**Related concepts:**

- [Job DSNTIJRT](#)

**Related reference:**

- [ROUTINES CREAT field](#)

### INSTALL SQL ID field

The INSTALL SQL ID field specifies the CURRENT SQLID setting that is to be used when SQL is processed by most Db2 installation and sample jobs.

The INSTALL SQL ID does not apply to the DSNTIJRT and DSNTIJRV jobs. For these jobs, use the ROUTINES CREAT and SEC DEF CREAT fields, to specify the CURRENT SQLID.

Acceptable values: 1 - 8 characters. The first character must be alphabetic.

Default: SYSADM

Update: option 6 on panel DSNTIPB
The installation CLIST assigns the value of this field as the value of CURRENT SQLID for installation jobs except DSNTIJRT and DSNTIJSR.

If you use the installation SYSOPR authority for Db2 installation, specify an installation system administrator ID. For Db2 migration, specify in the INSTALL PKG OWNER field an authorization ID with at least DBADM WITHOUT ACCESSCTRL authorization. Also, set the ROUTINES CREATOR, SEC DEF CREATOR, and INSTALL SQL ID fields to SYSINSTL.

**Related reference:**
- INSTALL PKG OWNER field
- SEC DEF CREATOR field
- ROUTINES CREATOR field
- INSTALL SQL ID field
- IVP PACKAGE OWNER field
- IVP SQL ID field

**INSTALL PKG OWNER field**
The INSTALL PKG OWNER field specifies the authorization ID to assign as the owner of packages and plans that are bound by most installation and sample jobs.

When the INSTALL PKG OWNER field is specified, it also sets the default owner of packages that are bound by the DSNTIJRT and DSNTIJSR jobs. When it is not specified, the default package owner is the authorization ID that is specified in the ROUTINES CREATOR field. You can specify the package ownership for individual routines on installation panels DSNTIPR through DSNTIPRT.

Acceptable values: 1 - 8 characters. The first character must be alphabetic.
- Default: blank
- Update: option 6 on panel DSNTIPB
- DSNZPxxx: none

The installation CLIST assigns the value of the INSTALL PKG OWNER field, which is the value of the OWNER parameter in BIND statements that are processed by installation jobs. If a value for this field is not specified, the OWNER clause is omitted from those BIND statements, except for those processed by jobs DSNTIJRT and DSNTIJSR. The INSTALL PKG OWN field is required when the ROUTINES CREATOR, SEC DEF CREATOR, and INSTALL SQL ID fields are set to SYSINSTL.

If you use installation SYSOPR authority for Db2 installation, specify an installation system administrator ID in the INSTALL PKG OWNER field.

If you use installation SYSOPR authority for migration, specify an authorization ID with at least DBADM WITHOUT ACCESSCTRL authority in the INSTALL PKG OWNER field.

When you specify a value in the INSTALL PKG OWNER field, you must also set the ROUTINES CREATOR, SEC DEF CREATOR, and INSTALL SQL ID fields to SYSINSTL.
**INSTALL GRANTEE(S) field**

The INSTALL GRANTEE(S) field specifies the authorization IDs that are to be granted access to objects that are created and bound by most installation and sample jobs.

This field does not apply to the DSNTIJRT and DSNTIJRV jobs. Use installation panels DSNTIPRA - DSNTIPRP to specify authorization IDs for routines that are provided by Db2. For more information, see [DSNTIPR1: Parameters for Db2-supplied routines panel](#).

Acceptable values: at least one valid authorization ID

Default: PUBLIC

Update: option 6 on panel DSNTIPB

DSNZPxxx: none

Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas. The authorization IDs that you enter in this field are granted the following privileges:

- The USE privilege for buffer pool BP0, the SYSDEFLT storage group, and table space DSNDB04.SYSDEFLT
- The CREATETAB and CREATETS privileges for the temporary database, DSNDB04
- The DELETE, INSERT, SELECT, and UPDATE privileges for tables and created global temporary tables that are created by installation jobs other than DSNTIJRT
- The EXECUTE privilege for packages and plans that are bound by installation jobs other than DSNTIJRT

PUBLIC grants the privileges to all users on the current server, including database requests using DRDA access. For regulatory compliance, do not use PUBLIC.

**IVP SQL ID field**

The IVP SQL ID field specifies the CURRENT SQLID setting when SQL is processed by most Db2 sample jobs.

Acceptable values: 1 - 8 characters, the first of which must be alphabetic

Default: SYSADM

Update: option 6 on panel DSNTIPB

DSNZPxxx: none

The installation CLIST assigns the value of the IVP SQL ID field, which is the CURRENT SQLID value for IVP jobs.

**Related tasks:**

[Verifying successful Db2 12 installation or migration with sample applications](#)

**IVP PACKAGE OWNER field**

The IVP PACKAGE OWNER field specifies the authorization ID to be assigned as the owner of packages and plans that are bound by Db2 sample jobs.

Acceptable values: 1 - 8 characters, the first of which must be alphabetic

Default: blank

Update: option 6 on panel DSNTIPB

DSNZPxxx: none
The installation CLIST assigns the value of the IVP PACKAGE OWNER field, which is the setting of the OWNER parameter in BIND statements processed by sample jobs. If this field is not specified, the OWNER clause is omitted from those BIND statements.

**Related tasks:**
- Verifying successful Db2 12 installation or migration with sample applications

### IVP GRANTEE(S) field

The IVP GRANTEE(S) field specifies the authorization IDs that are to be granted access to objects created and bound by Db2 sample jobs.

- **Acceptable values:** at least one valid authorization ID
- **Default:** PUBLIC
- **Update:** option 6 on panel DSNTIPB
- **DSNZPxxx:** none

Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas. The authorization IDs that you enter in this field are granted the following privileges:

- The USE privilege for STOGROUPs and table spaces that are created by IVP jobs
- The DBADM privilege for databases that are created by IVP jobs
- The DELETE, INSERT, SELECT, and UPDATE privileges for tables and created global temporary tables that are created by IVP jobs
- The EXECUTE privilege for packages and plans that are bound by IVP jobs
- The BIND privilege on most plans that are bound by IVP jobs

PUBLIC grants the privileges to all users on the current server, including database requests using DRDA access. For regulatory compliance, do not use PUBLIC.

**Related tasks:**
- Verifying successful Db2 12 installation or migration with sample applications

### PROVISION EXEC ID field

The PROVISION EXEC ID field specifies the default user ID for executing steps in a provisioning workflow.

This field is required, modifiable, and always selected as a z/OSMF variable when INSTALL TYPE ==> PROVISION is specified on panel DSNTIPA1 or DSNTILA1. Otherwise, this field is protected and always deselected as a z/OSMF variable.

- **Acceptable values:** 1 - 8 characters. The first character must be a letter.
- **Default:** SYSADM
- **Update:** none
- **DSNZPxxx:** none
- **z/OSMF variable:** AGEXECID

**Related tasks:**
- Creating artifacts for provisioning Db2 subsystems by running the installation CLIST

**Related reference:**
- DSNTIPA1 or DSNTILA1: Main panel
- INSTALL TYPE field
The entries on the DSNTIPG1 panel specify configuration options for writing image copies of the catalog and directory that are created by job DSNTIJIC to disk.

**Related concepts:**

- Related concepts: Copying catalog and directory objects (Db2 Utilities)

**Related tasks:**

- Creating an image copy of the Db2 directory and catalog: DSNTIJIC
- Migration step 5: Take image copies of the directory and catalog: DSNTIJIC
- Installation step 23: Back up the Db2 directory and catalog: DSNTIJIC

### INSTALL IC PREFIX field

The INSTALL IC PREFIX field specifies the prefix for image copy data sets that are created by Db2 installation and IVP jobs.

Acceptable values: 1 - 17 characters that form a valid z/OS data set name prefix

Default: DSN1210

Update: option 6 on panel DSNTIPB

DSNZPxxx: none

The installation CLIST assigns the value of this field as the prefix for all image copy data sets that are created by Db2 installation and IVP jobs.

### INSTALL DS PREFIX field

The INSTALL DS PREFIX field specifies the prefix for most data sets that are created by most Db2 installation and IVP jobs. This field does not apply to data sets that are created by job DSNTIJIN.

Acceptable values: 1 - 17 characters that form a valid z/OS data set name prefix

Default: DSN1210

Update: option 6 on panel DSNTIPB

DSNZPxxx: none
The installation CLIST assigns the value of this field as the prefix for most data sets that are created by Db2 installation and IVP jobs. However, the following exceptions apply:

- Image copy data sets use the prefix that is specified by the INSTALL IC PREFIX field.
- Db2 sample application libraries (prefix.DBRMLIB.DATA, prefix.RUNLIB.LOAD, and prefix.SRCLIB.DATA) use the prefix that is specified by the DATA SET NAME PREFIX field on panel DSNTIPA1.
- Db2 catalog and directory data sets and the Db2 administrative task scheduler file use the prefix that is specified by the CATALOG ALIAS field on panel DSNTIPA2.
- BSDS and active log copy data sets use the fields on panel DSNTIPH to specify the full name.

**IC DATA SET UNIT NAME field**

The IC DATA SET UNIT NAME field specifies the device type or unit name for allocating image copy data sets that are created by Db2 installation jobs.

<table>
<thead>
<tr>
<th>Acceptable values</th>
<th>Valid device type or unit name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>SYSDA</td>
</tr>
<tr>
<td>Update</td>
<td>option 7 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZPxxx</td>
<td>None</td>
</tr>
</tbody>
</table>

The installation CLIST assigns the value of this field as the device type for allocating image copies data sets created by Db2 installation jobs. By default, these jobs allocate image copies on disk. If the field specifies a tape device (TAPE, 3480, 3490, 3590), the installation CLIST converts these jobs to allocate the image copies on tape. Such conversion might increase the number of job steps to be processed.

In particular for installation job DSNTIJIC, the job for creating image copies of the Db2 catalog and directory the field value has the following effects:

- Specifying a tape device causes the installation CLIST to configure the DSNTIJIC job for serial copy processing and using JCL for allocating image copy data sets. Copy data sets are named prefix.IMAGCOPY.space-name, where prefix is the value specified in the INSTALL IC PREFIX field and space-name is the name of each table space in the Db2 catalog and directory.
- Specifying a disk device causes the installation CLIST to configure the DSNTIJIC job for parallel copy processing through Db2 utility LISTDEF statements and TEMPLATE statements for allocating image copy data sets. Copy data sets are named according to the value specified in the IC TEMPLATE EXPRESSION field.

**Related tasks:**

- Creating an image copy of the Db2 directory and catalog: DSNTIJIC (optional)

**Related reference:**

- [IC TEMPLATE EXPRESSION field](#)
- [Syntax and options of the LISTDEF control statement (Db2 Utilities)](#)
- [Sample LISTDEF control statements (Db2 Utilities)](#)
- [Syntax and options of the TEMPLATE control statement (Db2 Utilities)](#)
- [Sample TEMPLATE control statements (Db2 Utilities)](#)
IC TEMPLATE EXPRESSION field

The IC TEMPLATE EXPRESSION field specifies the Db2 TEMPLATE utility expression for allocating image copy data sets created by Db2 installation jobs.

- **Acceptable values:** Valid TEMPLATE utility control statement
- **Default:** DSNCAT.&DB..&SN..&IC..D&JU..T&TI.
- **Update:** option 7 on panel DSNTIPB
- **DSNZP:xx:** None

The installation CLIST assigns the value of this field as the TEMPLATE expression for allocating image copy data sets created by Db2 installation jobs. See the IC DATA SET UNIT NAME field for considerations regarding DSNTIJIC, the job for creating image copies of the Db2 catalog and directory.

**Related tasks:**
- Creating an image copy of the Db2 directory and catalog: DSNTIJIC (optional)

**Related reference:**
- [IC DATA SET UNIT NAME field](#)
- [Syntax and options of the TEMPLATE control statement (Db2 Utilities)](#)
- [Sample TEMPLATE control statements (Db2 Utilities)](#)

IC TEMPLATE DATA CLASS field

The IC TEMPLATE DATA CLASS field specifies the SMS data class name for allocating image copy data sets created by Db2 installation jobs.

- **Acceptable values:** Blank or a valid SMS data class name
- **Default:** Blank
- **Update:** option 7 on panel DSNTIPB
- **DSNZP:xx:** None

The installation CLIST assigns the value of this field, if specified, as the DATACLASS option of the TEMPLATE utility control statement for allocating image copy data sets created by DB2 installation jobs. If this field is blank, the DATACLASS option does not appear in the TEMPLATE utility control statement.

The installation CLIST does not use this field if the IC DATA SET UNIT NAME specifies a tape device.

**Related reference:**
- [Syntax and options of the TEMPLATE control statement (Db2 Utilities)](#)
- [Sample TEMPLATE control statements (Db2 Utilities)](#)

IC TEMPLATE MGMT CLASS field

The IC TEMPLATE MGMT CLASS field specifies the SMS management class name for allocating image copy data sets created by Db2 installation jobs.

- **Acceptable values:** Blank or a valid SMS management class name
- **Default:** Blank
- **Update:** option 7 on panel DSNTIPB
- **DSNZP:xx:** None

The installation CLIST assigns the value of this field, if specified, as the MGMTCLAS option of the TEMPLATE utility control statement for allocating
image copy data sets created by Db2 installation jobs. If this field is blank, the
MGMTCLAS option does not appear in the TEMPLATE utility control statement.

The installation CLIST does not use this field if the IC DATA SET UNIT NAME
specifies a tape device.

Related reference:

- Syntax and options of the TEMPLATE control statement (Db2 Utilities)
- Sample TEMPLATE control statements (Db2 Utilities)

**IC TEMPLATE STOR CLASS field**

The IC TEMPLATE STOR CLASS field specifies the SMS storage class name for
allocating image copy data sets created by Db2 installation jobs.

Acceptable values: Blank or a valid SMS data class name
Default: Blank
Update: option 7 on panel DSNTIPB
DSNZPxxx:

The installation CLIST assigns the value of this field, if specified, as the
STORCLAS option of the TEMPLATE utility control statement for allocating image
copy data sets created by Db2 installation jobs. If this field is blank, the
STORCLAS option does not appear in the TEMPLATE utility control statement.

The installation CLIST does not use this field if the IC DATA SET UNIT NAME
specifies a tape device.

Related reference:

- Syntax and options of the TEMPLATE control statement (Db2 Utilities)
- Sample TEMPLATE control statements (Db2 Utilities)

**DSNTIPH: System resource data set names panel**

The entries on the DSNTIPH panel specify the names of bootstrap data sets, active
logs, and archive logs. These entries also specify the number of copies (1 for single
logging or 2 for dual logging) for the active and archive logs.

Fields 1, 2, 4, 5, 7, and 8 on the DSNTIPH panel contain the prefix that was
entered in the BSDS AND ACTIVE LOG ICF CATALOG ALIAS field on
installation panel DSNTIPA4. If you scroll back to panel DSNTIPA4 and change the
BSDS AND ACTIVE LOG ICF CATALOG ALIAS value, the values for fields 1, 2, 4,
5, 7, and 8 on DSNTIPH change. When you scroll from panel DSNTIPA4 to panel
DSNTIPH, check these values and enter them again if necessary. In MIGRATE or
UPDATE modes, the BSDS AND ACTIVE LOG ICF CATALOG ALIAS value
cannot be changed, so the fields on DSNTIPH are not affected.

Dual logging improves reliability of recovering and, for active log reads, eases
device contention.

**Recommendation:** Specify dual logging for both active and archive logs. If you
specify dual logging, and an error occurs during offload to the archive logs, Db2
restarts the archive process using the second copy of the active log.
The value of the COPY 1 NAME field specifies the fully qualified name that is to be used for the first copy of the bootstrap data set.

Acceptable values: valid data set name; 1 - 33 characters
Default: DSNCA.T.BDS01 or DSNCA.T.DSN1.BDS01
Update: option 8 on panel DSNTIPB; not during migration

For non-data-sharing environments, the default prefix is DSNCA.T.BDSxx. For data sharing environments, the default prefix is DSNCA.T.DSN1.BDSxx. The resulting data set name is DSNCA.BDSxx.Annnnnn or DSNCA.T.DSN1.BDSxx.Annnnnn where:
• DSNCA is the value of the CATALOG ALIAS field on panel DSNTIPA2. You can change this portion of the data set prefix on this panel. If you change it, you need to supply another catalog alias. This additional catalog alias is not automatically defined by the installation process.
• DSN1 is the value of the MEMBER NAME field on panel DSNTIPK.
• xx is 01 for the first copy of the logs and 02 for the second copy.
• Annnnnnn is generated by Db2.

For the definition of a valid data set name, see [Valid data set name]

The value of the COPY 2 NAME field specifies the fully qualified name that is to be used for the second copy of the bootstrap data set.

Acceptable values: valid data set name; 1 - 33 characters
Default: DSNCA.T.BDS02 or DSNCA.T.DSN1.BDS02
Update: option 8 on panel DSNTIPB; not during migration

For non-data-sharing environments, the default prefix is DSNCA.T.BDSxx. For data sharing environments, the default prefix is DSNCA.T.DSN1.BDSxx. The resulting data set name is DSNCA.T.BDSxx.Annnnnn or DSNCA.T.DSN1.BDSxx.Annnnnn where:
• DSNCA is the value of the CATALOG ALIAS field on panel DSNTIPA2. You can change this portion of the data set prefix on this panel. If you change it, you need to supply another catalog alias. This additional catalog alias is not automatically defined by the installation process.
• DSN1 is the value of the MEMBER NAME field on panel DSNTIPK.
• xx is 01 for the first copy of the logs and 02 for the second copy.
• Annnnnnn is generated by Db2.
For non-data-sharing environments, the default prefix is DSNCAT.BSDSxx. For data sharing environments, the default prefix is DSNCAT.DSN1.BSDSxx. The resulting data set name is DSNCAT.BSDSxx.<nnnnnnnnnnnnnnnnnnn> or DSNCAT.DSN1.BSDSxx.<nnnnnnnnnnnnnnnnnnn>
where:

- DSNCAT is the value of the CATALOG ALIAS field on panel DSNTIPA2. You can change this portion of the data set prefix on this panel. If you change it, you need to supply another catalog alias. This additional catalog alias is not automatically defined by the installation process.
- DSN1 is the value of the MEMBER NAME field on panel DSNTIPK.
- xx is 01 for the first copy of the logs and 02 for the second copy.
- <nnnnnnnnnnnnnnnnnnn> is generated by Db2.

For the definition of a valid data set name, see "Valid data set name".

**NUMBER OF COPIES field (TWOACTV subsystem parameter)**

The TWOACTV subsystem parameter specifies the number of copies of the active log that Db2 is to maintain: 1 (single logging) or 2 (dual logging).

Acceptable values: 1, 2
Default: 2
Update: option 8 on panel DSNTIPB; not during migration
DSNZPxxx: DSN6LOGP TWOACTV

Dual logging increases reliability of recovery. If your Db2 subsystem creates copies of the archive log on tape, two tape drives must be available during the offload process.

**COPY 1 PREFIX field**

The value of the COPY 1 PREFIX field specifies the prefix that is to be used for the first copy of the active log data sets.

Acceptable values: valid data set name prefix; 1 - 30 characters
Default: DSNCAT.LOGCOPY1 or DSNCAT.DSN1.LOGCOPY1
Update: option 8 on panel DSNTIPB; not during migration
DSNZPxxx: none

For non-data-sharing environments, the default prefix is DSNCAT.LOGCOPYx. For data sharing environments, the default prefix is DSNCAT.DSN1.LOGCOPYx. The resulting data set name is DSNCAT.LOGCOPYx.<nnnnnnnnnnnnnnnnnnn> or DSNCAT.DSN1.LOGCOPYx.<nnnnnnnnnnnnnnnnnnn>, where:

- DSNCAT is the value of the CATALOG ALIAS field on panel DSNTIPA2. You can change this portion of the data set prefix on this panel. If you change it, you need to specify another catalog alias. This additional catalog alias is not automatically defined by the installation process.
- DSN1 is the of the MEMBER NAME field on panel DSNTIPK.
- LOGCOPY is part of the data set prefix that you can change on this panel.
- x is 1 for the first copy of the logs and 2 for the second copy.
- <nnnnnnnnnnnnnnnnnnn> is the data set number.

For the definition of a valid data set name, see "Valid data set name".

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST  111
COPY 2 PREFIX field

The value of the COPY 2 PREFIX field specifies the prefix that is to be used for the second copy of the active log data sets.

Acceptable values: valid data set name prefix; 1 - 30 characters
Default: DSNCAT.LOGCOPY2 or DSNCAT.DSN1.LOGCOPY2
Update: option 8 on panel DSNTIPB; not during migration
DSNZP:xxx: none

If you are using single logging, accept the default value. Do not leave the entry blank.

For non-data-sharing environments, the default prefix is DSNCAT.LOGCOPYx. For data sharing environments, the default prefix is DSNCAT.DSN1.LOGCOPYx. The resulting data set name is DSNCAT.LOGCOPYx.Annnnnn or DSNCAT.DSN1.LOGCOPYx.Annnnnn, where:

- DSNCAT is the value of the CATALOG ALIAS field on panel DSNTIPA2. You can change this portion of the data set prefix on this panel. If you change it, you need to specify another catalog alias. This additional catalog alias is not automatically defined by the installation process.
- DSN1 is the value of the MEMBER NAME field on panel DSNTIPK.
- LOGCOPY is part of the data set prefix that you can change on this panel.
- x is 1 for the first copy of the logs and 2 for the second copy.
- nnnnnnn is the data set number.

For the definition of a valid data set name, see Valid data set name.

NUMBER OF COPIES field (TWOARCH subsystem parameter)

The TWOARCH subsystem parameter specifies the number of copies of the archive log that Db2 is to produce during offloading.

Acceptable values: 1, 2
Default: 2
Update: option 8 on panel DSNTIPB
DSNZP:xxx: DSN6LOGP TWOARCH

If you run the installation or migration CLIST, the setting of the NUMBER OF COPIES field determines the setting of the TWOARCH subsystem parameter. Valid values of the TWOARCH subsystem parameter are NO and YES.

1 Db2 produces a single copy of the archive log during offloading. This setting corresponds with a value of NO for the TWOARCH subsystem parameter.

2 Db2 produces two copies of the archive log during offloading. Dual logging increases reliability of recovery. This setting corresponds with a value of YES for the TWOARCH subsystem parameter.

COPY 1 PREFIX field (ARCPFX1 subsystem parameter)

The ARCPFX1 subsystem parameter specifies the prefix that is to be used for the first copy of the archive log data set.

Acceptable values: valid data set name prefix; 1 - 35 characters
Default: DSNCAT.ARCHLOG1 or DSNCAT.DSN1.ARCLG1
For definitions of valid data set names, see [Valid data set name]

**COPY 2 PREFIX field (ARCPFX2 subsystem parameter)**
The ARCPFX2 subsystem parameter specifies the prefix that is to be used for the second copy of the archive log data set. If you use single logging, accept the default value.

Acceptable values: valid data set name prefix; 1 - 35 characters
Default: DSNCAT.ARCHLOG2 or DSNCAT.DSN1.ARCLG2
Update: option 8 on panel DSNTIPB
DSNZP.xxx: DSN6ARVP ARCPFX2

Do not leave the COPY 2 PREFIX field blank.

**TIMESTAMP ARCHIVES field (TSTAMP subsystem parameter)**
The TSTAMP subsystem parameter specifies whether the Db2 archive log data set name is to contain the date and time that the archive log data set was created.

Acceptable values: NO, YES, EXT
Default: NO
Update: option 8 on panel DSNTIPB
DSNZP.xxx: DSN6ARVP TSTAMP

**NO** The archive data set name does not contain a timestamp.

**YES**

The maximum allowable length of the user-controlled portion of the archive log prefix is reduced from 35 characters to 19 characters. This reduction in size permits the 16-character date and time qualifiers (timestamp) to be added to the archive log data set prefix. The timestamp format is as follows:

.Dyyddd.Thhmmssst,

where:
- D is the letter D.
- yy is the last two digits of the year.
- ddd is the day of the year.
- T is the letter T.
- hh is the hour.
- mm is the minutes.
- ss is the seconds.
- t is tenths of a second.

**EXT**
The archive data set name contains a timestamp with an extended date component in the format:

.Dyyyyyddd.

A value of EXT in this field causes the lengths of the values that are entered for field COPY 1 PREFIX and field COPY 2 PREFIX to be audited to ensure that neither exceeds 17 bytes (19 bytes for other settings of TIMESTAMP ARCHIVES).
DSNTIPT: Db2 data set names panel

The entries on the DSNTIPT panel establish data set names for the libraries that are used in the Db2 CLIST and JCL that Db2 provides. The values that you enter on this panel are edited into all pertinent sample and installation jobs.

You can enter these values in one of three ways: same data set name prefix, no data set name prefix, or a new data set name prefix. The following table summarizes these selections.

<table>
<thead>
<tr>
<th>If you use ...</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same data set name prefix or data set names</td>
<td>Current data sets are deleted and reallocated for installation and migration.</td>
</tr>
<tr>
<td>No data set names</td>
<td>No new output is created. Previous output remains intact.</td>
</tr>
<tr>
<td>New prefix</td>
<td>Output is saved in new data set. Previous output remains intact.</td>
</tr>
</tbody>
</table>

When you run the CLIST in INSTALL or MIGRATE mode, the following warning message is displayed for any output data set that already exists:

DSNT434I WARNING, DATA SETS MARKED WITH ASTERISKS EXIST AND WILL BE OVERWRITTEN

To avoid deleting these data sets, take one of the following actions:

- Press Enter to leave the installation process.
- Change the data set names.

Press Enter again if you want to continue; this overwrites your data sets.

When you run the CLIST in UPDATE mode, this panel is displayed immediately after panel DSNTIPA1. This allows you to check the SDSNSAMP data set name to see if it is the one you want to use for the DSNTIJUZ job. Data sets are marked with an asterisk but not deleted or reallocated if you use the same name. Instead, the data set is compressed, and only the DSNTIJUZ member is replaced within the data set. Other members in the data set are left unchanged. When the CLIST completes processing and returns to panel DSNTIPA1, the following message is displayed:

WARNING MESSAGE(S) WAS PRODUCED DURING CLIST TAILORING SESSION
TEMP CLIST LIBRARY and SAMPLE LIBRARY are data sets that are allocated by the installation CLIST for edited output. CLIST LIBRARY is allocated by DSNTIJVC. If the INPUT MEMBER NAME field on installation panel DSNTIPA1 is set to DSNTIIXA or DSNTIDXX, the three data sets default to prefix.NEW.SDSNTEMP, prefix.NEW.SDSNCLST, and prefix.NEW.SDSNSAMP respectively. In these names, prefix is the value that is entered for the DATA NAME PREFIX field on installation panel DSNTIPA1. The following table shows the job-tailoring fields.

**Table 21. Job-tailoring fields**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Tailored output</th>
<th>No tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing</td>
<td>All three fields entered</td>
<td>All three fields blank</td>
</tr>
<tr>
<td>Migrating</td>
<td>All three fields entered</td>
<td>All three fields blank</td>
</tr>
<tr>
<td>Updating</td>
<td>SAMPLE LIBRARY entered</td>
<td>SAMPLE LIBRARY blank</td>
</tr>
</tbody>
</table>

Db2 adds blanks to these fields after a successful tailoring session to avoid writing over the tailored output.

**TEMP CLIST LIBRARY field**

The value of the TEMP CLIST LIBRARY field specifies the name of the data set where edited CLISTS are to be placed.

- **Acceptable values:** valid data set name: see Valid data set name
- **Default:** prefix.NEW.SDSNTEMP
- **Update:** cannot change during update
- **DSNZPxxx:** none

This field must not be blank if you are tailoring output.

**SAMPLE LIBRARY field**

The value of the SAMPLE LIBRARY field specifies the name of the edited JCL library that is to be used by sample and installation jobs.

- **Acceptable values:** valid data set name: see Valid data set name

**Figure 13. Db2 data set names panel: DSNTIPT**
In update mode, the new sample library data set is not reallocated. It is compressed and member DSNTIJUZ is overwritten. This field must not be blank if you are tailoring output.

**CLIST LIBRARY field**

The CLIST LIBRARY field identifies the name of the data set into which job DSNTIJVC is to load all CLISTs.

Acceptable values: valid data set name: see [Valid data set name]

Default: prefix.NEW.SDSNCLST

Update: cannot change during update

DSNZPxxx: none

This field must not be blank if you are tailoring output.

**APPLICATION DBRM field**

The value of the APPLICATION DBRM field specifies the name of the library that is to be used for Db2 sample application DBRMs.

Acceptable values: valid data set name: see [Valid data set name]

Default: prefix.DBRMLIB.DATA.suffix

Update: cannot change during update

DSNZPxxx: none

The APPLICATION DBRM field, APPLICATION LOAD field, and DECLARATION LIBRARY field are for Db2-provided sample applications. The names of your own development libraries are most likely different from the default values that are shown here. Job DSNTIJMV references another set of DBRMLIB, RUNLIB, and SRCLIB data sets for SYS1.PROCLIB. See for more information. These fields must not be blank.

**APPLICATION LOAD field**

The value of the APPLICATION LOAD field specifies the name of the Db2 sample application load module library.

Acceptable values: valid data set name: see [Valid data set name]

Default: prefix.RUNLIB.LOAD.suffix

Update: cannot change during update

DSNZPxxx: none

The APPLICATION DBRM field, APPLICATION LOAD field, and DECLARATION LIBRARY field are for Db2-provided sample applications. The names of your own development libraries most likely are different from the default values that are shown here. Job DSNTIJMV references another set of DBRMLIB, RUNLIB, and SRCLIB data sets for SYS1.PROCLIB. See for more information. These fields must not be blank.
DECLARATION LIBRARY field
The value of the DECLARATION LIBRARY field specifies the name of the Db2 declaration library that is to be used for sample application include files.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SRCLIB.DATA.suffix
Update: cannot change during update
DSNZPxxx: none

The DECLARATION LIBRARY field is for Db2-provided sample applications. The name of your own development library is most likely different from the default value for this field.

LINK LIST LIBRARY field
The LINK LIST LIBRARY field specifies the name of the APF-authorized Db2 early code library.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDSNLINK.suffix
Update: cannot change during update
DSNZPxxx: none

The LINK LIST LIBRARY field specifies the name of a data set that is allocated during SMP processing. This field must not be blank.

LOAD LIBRARY field
The value of the LOAD LIBRARY field specifies the name of the main APF-authorized Db2 load module library that is to be used by installation and sample jobs.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDSNLOAD.suffix
Update: cannot change during update
DSNZPxxx: none

The LOAD LIBRARY field specifies the name of a data set that is allocated during SMP processing. This field must not be blank.

JDBC AND SQLJ DLLS field
The JDBC AND SQLJ DLLS field specifies the name of the main DLL library for Db2 Java support. This library is used by installation and sample jobs and some of the stored procedures that are provided by Db2.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDSNLOD2.suffix
Update: cannot change during update
DSNZPxxx: none

The JDBC AND SQLJ DLLS field specifies the name of a data set that is allocated during SMP processing. This field cannot be blank.

MACRO LIBRARY field
The value of the MACRO LIBRARY field specifies the name of the macro library that is to be used by installation and sample jobs. This library contains the CICS
and IMS attachment facility macros, the initialization parameter macros, and some
data-mapping macros that are needed for some applications.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDSNMACS.suffix
Update: cannot change during update
DSNZPxxx: none

The MACRO LIBRARY field specifies the name of a data set that is allocated
during SMP processing. This field must not be blank.

EXIT LIBRARY field
The value of the EXIT LIBRARY field specifies the name of the library where your
DSNZPxxx module, application defaults load module (dsnldcexp), and exit routines
are to be placed.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDSNEXIT.suffix
Update: cannot change during update
DSNZPxxx: none

When you use prefix.SDSNLOAD and prefix.SDSNEXIT together, list
prefix.SDSNEXIT first to override the IBM defaults in prefix.SDSNLOAD.

The EXIT LIBRARY field specifies the name of a data set that is allocated during
SMP processing. This field must not be blank.

DBRM LIBRARY field
The value of the DBRM LIBRARY field identifies the name of the library where the
DBRMs that are shipped with Db2 are to be placed.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDSNDBRM.suffix
Update: cannot change during update
DSNZPxxx: none

The DBRM LIBRARY field specifies the names of data sets that are allocated
during SMP processing. This field must not be blank.

IRLM LOAD LIBRARY field
The value of the IRLM LOAD LIBRARY field specifies the name of the IRLM load
library data set that is to be used in the IRLM procedure.

Acceptable values: valid data set name: see Valid data set name
Default: prefix.SDXRRESL.suffix
Update: cannot change during update
DSNZPxxx: none

This field must not be blank. It specifies the name of a data set that is allocated
during SMP processing.
**IVP DATA LIBRARY field**

The value of the IVP DATA LIBRARY field specifies the data set name of SDSNIVPD. SDSNIVPD is the SMP/E target library for the Db2 installation verification procedure (IVP) input data and for the expected output from the sample applications.

Acceptable values: valid data set name; see Valid data set name

Default: prefix.SDSNIVPD.suffix

Update: cannot change during update

DSNZPxxx: none

**INCLUDE LIBRARY field**

The value of the INCLUDE LIBRARY field identifies the name of the include library data set that is to be used by sample and installation jobs.

Acceptable values: valid data set name; see Valid data set name

Default: prefix.SDSNC.H

Update: cannot change during update

DSNZPxxx: none

This library is used in the Db2 language PROCS for C and C++.

**DSNTIPU: Host language data set names panel 1**

The entries on panels DSNTIPU, DSNTIPU1, and DSNTIPW establish data set names for other product libraries. The values that you enter on these panels are edited into sample and installation jobs. If you do not have the product, accept the default values. Jobs for those particular products should not be run.

Db2 makes assumptions about which one of the possible C, C++, and PL/I compilers that you are using, depending on the values you supply or leave as default in the C, C++, and PL/I fields.

Many data set names for other products appear in the jobs. You can enter most of these data sets on this panel and on the next two installation panels, DSNTIPU1 and DSNTIPW. These names are shown in the following table as they appear in the jobs that are shipped with Db2. Change the names of the data sets if they are different at your site.

**Table 22. Data set names that are used in jobs for related products**

<table>
<thead>
<tr>
<th>Job</th>
<th>Data set name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTEJ1</td>
<td>SYS1.MACLIB</td>
<td>Assembly macro library</td>
</tr>
<tr>
<td></td>
<td>SYS1.SORTLIB</td>
<td>DFSORT load modules (can be deleted if DFSORT is in link list)</td>
</tr>
<tr>
<td>DSNTEJ1L</td>
<td>CEE.SCEELKED</td>
<td>Language Environment linkage editor library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ1P</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ1U</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td></td>
<td>SYS1.SORTLIB</td>
<td>DFSORT load modules (can be deleted if DFSORT is in link list)</td>
</tr>
<tr>
<td>DSNTEJ2A</td>
<td>SYS1.SORTLIB</td>
<td>DFSORT load modules (can be deleted if DFSORT is in link list)</td>
</tr>
<tr>
<td>DSNTEJ2C</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ2D</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ2E</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>Job</td>
<td>Data set name</td>
<td>Function</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>DSNTEJ2F</td>
<td>SYS1.MACLIB</td>
<td>Assembler macro library</td>
</tr>
<tr>
<td></td>
<td>SYS1.VSF2FORT</td>
<td>VS Fortran run time library</td>
</tr>
<tr>
<td>DSNTEJ2P</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ3C</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ3P</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ4C</td>
<td>IMSVS.RESLIB</td>
<td>IMS linkage editor library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ4P</td>
<td>IMSVS.RESLIB</td>
<td>IMS linkage editor library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEELKED</td>
<td>PL/I linkage editor base library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEERUN</td>
<td>PL/I dynamic run time base library</td>
</tr>
<tr>
<td>DSNTEJ5A</td>
<td>CICS410.SDFHLOAD</td>
<td>CICS command translator and linkage editor</td>
</tr>
<tr>
<td></td>
<td>CICS410.SDFHMAC</td>
<td>CICS macro library</td>
</tr>
<tr>
<td></td>
<td>SYS1.MACLIB</td>
<td>Assembler macro library</td>
</tr>
<tr>
<td>DSNTEJ5C</td>
<td>CICS410.SDFHLOAD</td>
<td>CICS command translator and linkage editor</td>
</tr>
<tr>
<td></td>
<td>IGY:SIGYCOMP</td>
<td>Enterprise COBOL for z/OS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See also the list of libraries that are used by DSNH CLIST in [DSNTEJ2F (TSO CLIST) (Db2 Commands)]</td>
</tr>
<tr>
<td>DSNTEJ5P</td>
<td>CICS410.SDFHLOAD</td>
<td>CICS command translator and linkage editor</td>
</tr>
<tr>
<td></td>
<td>CICS410.SDFHPLI</td>
<td>CICS PL/I linkage editor library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEELKED</td>
<td>Language Environment link editor library</td>
</tr>
<tr>
<td>DSNTEJ6D</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6P</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6R</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEEH.H</td>
<td>C library headers</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEELKED</td>
<td>Language Environment linkage editor library</td>
</tr>
<tr>
<td>DSNTEJ6S</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6T</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6U</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6V</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6W</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ6Z</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ61</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ62</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ63</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ64</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ65</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ66</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ7</td>
<td>SYS1.SORTLIB</td>
<td>DFSORT load modules (can be deleted if DFSORT is in link list)</td>
</tr>
<tr>
<td>DSNTEJ71</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ73</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ75</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ76</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ77</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJ78</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTEJXP</td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td>DSNTIJMV</td>
<td>SYS1.MACLIB</td>
<td>Assembler macro library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEERUN</td>
<td>Language Environment dynamic run time library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEELKED</td>
<td>Language Environment linkage editor library</td>
</tr>
<tr>
<td></td>
<td>EDCPRLK</td>
<td>Language Environment pre-link editor library</td>
</tr>
<tr>
<td></td>
<td>CEE.SCEEMSGP</td>
<td>Language Environment pre-link message file</td>
</tr>
<tr>
<td></td>
<td>CBC.SCCNCMP</td>
<td>C/C++ for z/OS compiler library</td>
</tr>
<tr>
<td></td>
<td>CCNDRVR</td>
<td>C/C++ compiler load module</td>
</tr>
</tbody>
</table>
### Table 22. Data set names that are used in jobs for related products (continued)

<table>
<thead>
<tr>
<th>Job</th>
<th>Data set name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEE.SCEEH.H</td>
<td></td>
<td>C library headers</td>
</tr>
<tr>
<td>CEE.SCLBH.H</td>
<td></td>
<td>C++ library headers</td>
</tr>
<tr>
<td>CEE.SCEECPP</td>
<td></td>
<td>C++ autolink library</td>
</tr>
<tr>
<td>CBC.SCLBCPP</td>
<td></td>
<td>C++ class library</td>
</tr>
<tr>
<td>CICS.SCLBCPP</td>
<td></td>
<td>CICS library for COBOL</td>
</tr>
<tr>
<td>CICS.SDFHLOAD</td>
<td></td>
<td>CICS command translator and linkage editor</td>
</tr>
<tr>
<td>CICS.SDFLPLI</td>
<td></td>
<td>CICS library for PL/I</td>
</tr>
<tr>
<td>prefix.SDSNLOAD(DSNHPC)</td>
<td></td>
<td>Db2 precompiler</td>
</tr>
<tr>
<td>prefix.SDSNLOAD</td>
<td></td>
<td>Db2 linkage editor library</td>
</tr>
<tr>
<td>DSNHPC</td>
<td></td>
<td>Db2 precompiler module</td>
</tr>
<tr>
<td>GDDM.SADMSAM</td>
<td></td>
<td>GDDM macro library</td>
</tr>
<tr>
<td>GDDM.SADMMOD</td>
<td></td>
<td>GDDM load module library</td>
</tr>
<tr>
<td>IGY.SIGYCOMP</td>
<td></td>
<td>Enterprise COBOL for z/OS compiler library</td>
</tr>
<tr>
<td>IGYCRCTL</td>
<td></td>
<td>Enterprise COBOL for z/OS compiler load module</td>
</tr>
<tr>
<td>IMSVS.RESLIB</td>
<td></td>
<td>IMS linkage editor library</td>
</tr>
<tr>
<td>ISP'SISLOAD</td>
<td></td>
<td>ISPF ISPLINK module</td>
</tr>
<tr>
<td>IBM.SIBMZCMP</td>
<td></td>
<td>Enterprise PL/I for z/OS compiler library</td>
</tr>
<tr>
<td>SYS1.VSF2FORT</td>
<td></td>
<td>VS Fortran run time library</td>
</tr>
</tbody>
</table>

Use this panel to define the data set names of your IBM Language Environment, C/370™, or C/C++ program product libraries. For more information about these libraries, consult the appropriate program product documentation. For more information about these libraries, consult the appropriate program product documentation.

Data sets that are specified on this panel are used by the Db2 installation process to tailor the Db2 language procedures that are generated by installation job DSNTIJMV:

- DSNHASM can be used to prepare Db2 programs using assembly language
- DSNHC can be used to prepare Db2 programs that use C.
- DSNHCPP can be used to prepare Db2 programs that use C++.
- DSNHCPP2 can be used to prepare a class and a client for a Db2 object-oriented program that use C++.
- DSNHCPPS contains the header file search path that is to be used by DSNHCPP and DSNHCPP2.
- DSNHFOR can be used to prepare Db2 programs using FORTRAN
- DSNHICOB can be used to prepare Db2 programs using COBOL
- DSNHPLI can be used to prepare Db2 programs using PL/I
- DSNHSQNL can be used to prepare Db2 external SQL procedures

Data sets that you specify on this panel are also used by the Db2 installation process to tailor the Db2 Interactive (DB2I) program preparation CLIST, DSNH.

When the compiler fields are left blank, the DSNH CLIST and the provided JCL procedures operate differently. The DSNH CLIST issues a specific call statement, using the default load module data set name as the argument of the call. The JCL procedures use the z/OS link list to find the data set in which the load module resides.
Use fields 1 through 4 of DSNTIPU to specify the IBM Language Environment run

time environment, link editor, and pre-link editor message libraries. The CLIST

assumes that these libraries are used by all language products except FORTRAN.

Use fields 6 through 15 on panel DSNTIPU to define C/370 Version 2 Release 1 (C

only), AD/Cycle, C/370 Version 1 Release 2 (C only), C/C++ for MVS/ESA

Version 3 Release 2, C/C++ for OS/390®, or C/C++ for z/OS.Use fields 6 through

13 to define C/370 Version 2 Release 1 (C only), AD/Cycle, C/370 Version 1

Release 2 (C only), C/C++ for MVS/ESA Version 3 Release 2, C/C++ for OS/390,
or C/C++ for z/OS. If you need to define C++, you must define C/C++ for C as

well.

If C is not installed on your system:

• Accept the default values for fields 6 through 15 of DSNTIPU. Accept the default

values for fields 6 through 13.

• Do not run IVP jobs DSNTEJ2D, DSNTEJ2U, DSNTEJ6D, DSNTEJ6R, DSNTEJ6T,

DSNTEJ63, DSNTEJ6W, DSNTEJ6Z, DSNTEJ64, DSNTEJ65, DSNTEJ66,

DSNTEJ71, DSNTEJ73, and DSNTEJ75.

If C++ is not installed on your system:

• Accept the default values for fields 6 through 15 of DSNTIPU. Accept the default

values for fields 6 through 13.

• Do not run jobs DSNTEJ2E or DSNTEJ6V. Skip steps PH02US08 and PH02US09

of IVP job DSNTEJ2U.

• Remove all statements that refer to DAYNAME and MONTHNAME from part

DSNTESU in the prefix:SDSNSAMP library if C++ is not available.

If COBOL is not installed on your system:

• Accept the default value for fields 1 and 2 of DSNTIPU1. Accept the default

value for field 14.

• Do not run IVP jobs DSNTEJ2C, DSNTEJ3C, DSNTEJ4C, DSNTEJ5C, DSNTEJ61,

DSNTEJ62, DSNTEJ66, DSNTEJ76, DSNTEJ77, and DSNTEJ78.

If Fortran is not installed on your system:

• Accept the default values for fields 3 and 4 of DSNTIPU1. Accept the default

values for fields 15 and 16.

• Do not run IVP job DSNTEJ2F.

If PL/I is not installed on your system:

• Accept the default values for fields 5 and 6 of DSNTIPU1. Accept the default

values for field 17.

• Do not run IVP jobs DSNTEJ1P, DSNTEJ2P, DSNTEJ3P, DSNTEJ4P, DSNTEJ5P,

DSNTEJ6P, DSNTEJ6S, and DSNTEJ6U.
### IBM LE RUNTIME LIBRARY field

The IBM LE RUNTIME LIBRARY field identifies the name of the IBM Language Environment dynamic run time library that is to be used by sample and installation jobs.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>blank, or valid data set name: see <a href="#">Valid data set name</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>CEE.SCEERUN</td>
</tr>
<tr>
<td>Update:</td>
<td>cannot change during update</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>none</td>
</tr>
</tbody>
</table>

The data set name typically includes the qualifier SCEERUN. Leave this field blank if SCEERUN is in the link list. If you enter a value in this field, it is used in the STEPLIB concatenation of the JCL procedures generated by installation job DSNTIJMV, including the following procedures:

- The DSNDBM1 and DSNDIST address spaces for Db2
- The Db2–supplied WLM procedures DSNWLM and DSNCICS
- The Db2 language procedures
- The JOBLIB concatenation of many IVP jobs

If you plan to use Db2 to run XPLINK or AMODE 64 applications, provide the SCEERUN and SCEERUN2 libraries for IBM Language Environment in the z/OS program search order. See [Description of Language Environment target libraries](#) for more information.

### IBM LE RUNTIME LIBRARY 2 field

The IBM LE RUNTIME LIBRARY 2 field identifies the name of the IBM Language Environment dynamic run time library for programs that require the library to be in an extended partitioned data set (PDSE).

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>blank, or valid data set name: see <a href="#">Valid data set name</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>CEE.SCEERUN2</td>
</tr>
<tr>
<td>Update:</td>
<td>cannot change during update</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>none</td>
</tr>
</tbody>
</table>
The data set name typically includes the qualifier SCEERUN2. Leave this field blank if SCEERUN2 is in the link list. If you enter a value in this field, it is used in the STEPLIB concatenation of the DSNDBM1 JCL procedure that is generated by installation job DSNTIJMV.

If you plan to use Db2 to run XPLINK or AMODE 64 applications, provide the SCEERUN and SCEERUN2 libraries for IBM Language Environment in the z/OS program search order. See Description of Language Environment target libraries (z/OS Language Environment Customization) for more information.

**IBM LE LINK EDIT LIB field**
The IBM LE LINK EDIT LIB field identifies the name of the IBM Language Environment linkage editor library that sample and installation jobs are to use.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CEE.SCEELKED
Update: cannot change during update
DSNZPxxx: none

The data set name typically includes the qualifier SCEELKED. SCEELKED is required to link-edit load modules for C, C++, COBOL, and PL/I application programs. The value that you enter here is used to customize the Db2 language procedures and the DSNH CLIST.

**IBM LE PRELINK MSG LIB field**
The IBM LE PRELINK MSG LIB field identifies the data set name for messages that are to be issued by the IBM Language Environment pre-linkage editor (EDCPLRLK).

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CEE.SCEEMSGP
Update: cannot change during update
DSNZPxxx: none

The SMP/E target data set name typically includes the qualifier SCEEMSGP. SCEEMSGP is required to pre-link edit load modules for C, C++, COBOL, and PL/I application programs. The value that you enter here is used to customize Db2 language procedures and the DSNH CLIST.

**HIGH LEVEL ASSEMBLER LIB field**
The value of the HIGH LEVEL ASSEMBLER LIB field specifies the data set name of the assembler load module library. This value is to be used by sample and installation jobs.

Acceptable values: blank, or valid data set name: see [Valid data set name](#)
Default: none
Update: cannot change during update
DSNZPxxx: none

The data set name typically includes the qualifier SASMMOD1. The value that you specify for this field is used to customize the DSNHASM language procedure and the DSNH CLIST. It is also added to the STEPLIB concatenation of each Db2-provided job that uses the assembler. You can leave this field blank if the library is in the link list.
C/CPP COMPILER MODULE field

The C/CPP COMPILER MODULE field specifies the load module name of the C/370 or C/C++ compiler that is to be used on your system.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: CCNDRVR
Update: cannot change during update
DSNZPxxx: none

The default value is CCNDRVR for C/C++ for z/OS. The value that you enter here is used to determine the configuration for the DSNHC, DSNHCPP, DSNHCPP2, and DSNHSQL language procedures, the DSNH CLIST, and the IVP jobs that use C and C++. The value is used as follows:

- If the entry begins with the string “EDC”, the CLIST configures your system to use C/370.

Important: You cannot use C/370 and C++.

- If the entry begins with the string “CBC”, the CLIST configures your system to use C/C++ for MVS/ESA Version 3 Release 2.

- If the entry begins with any other value, including blanks, the CLIST configures your system to use C/C++ for z/OS.

C/CPP COMPILER LIBRARY field

The C/CPP COMPILER LIBRARY field specifies the name of the compiler library that is to be used for C/C++ or C/370.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: CEE.SCCNCMP
Update: cannot change during update
DSNZPxxx: none

This value is used to customize the DSNHC, DSNHCPP, DSNHCPP2, and DSNHSQL language procedures, and the DSNH CLIST. This field can be left blank if the compiler library is in the link list.

The following guidelines apply to the target data set name:

- For C/C++ for z/OS, the SMP/E target data set name typically includes the qualifier SCCNCMP.
- For C/C++ for MVS/ESA Version 3 Release 2, the SMP/E target data set name typically includes the qualifier SCBC3CMP.
- For C/370, the SMP/E target data set name typically includes the qualifier SEDCDCMP or SEDCCCOMP.

C/CPP HEADER LIBRARY field

The C/CPP HEADER LIBRARY field specifies the name of the header include library that is to be used for C/C++ or C/370.

Acceptable values: blank, or valid load module name: see Valid data set name
Default: CEE.SCEEH.H
Update: cannot change during update
DSNZPxxx: none
• For C/C++, the SMP/E target data set name typically includes the qualifier SCEEH.H.

• For C/370, the SMP/E target data set name typically includes the qualifier SEDCDCMP or SEDCCOMP.

This value is used to customize the DSNHC, DSNHCPP, DSNHCPP2, and DSNHSQNL language procedures, and the DSNH CLIST. This field can be left blank if the compiler library is in the link list.

**C/370 COMPILER MESSAGES field**
The value of the C/370 COMPILER MESSAGES field is the name of the message library that is to be used for the C/370 compiler.

Acceptable values: blank, or valid data set name: see [Valid data set name](#)
Default: none
Update: cannot change during update
DSNZPxxx: none

The data set name typically includes the qualifier SEDCDMSG or SEDCMSGS. If you are using C or C++, leave this field blank. This value is used to customize the DSNHC and DSNHSQNL language procedures and the DSNH CLIST.

You can specify C/C++ for MVS/ESA V3R1, or C/C++ for MVS/ESA V3R2. These are shipped as single products, but you need to define them separately on this line and on line 1, depending on your need for C or C++. If you specify a name in this field, a STEPLIB is added to the compiler step of the DSNHCPP and DSNHCPP2 procedures in job DSNTIJMV, and to the C++ portion of the DSNH CLIST. You can leave this field blank if the compiler library is in the link list.

**C LE PRELINKER PHASE field**
The value of the C LE PRELINKER PHASE field indicates whether to customize the DSNHC language procedure to include a procedure step that executes the IBM Language Environment prelinker.

Acceptable values: NO or YES
Default: NO
Update: cannot change during update
DSNZPxxx: none

**NO** Indicates that DSNHC is customized to prepare C programs without using the prelinker. Object code is passed directly from the compiler to the z/OS binder to create a program object, which must reside in a partitioned data set extended (PDSE). NO is the default setting.

**YES** Indicates that DSNHC is customized to prepare C programs using the prelinker. Object code is passed from the compiler to the IBM Language Environment Prelinker and then to the z/OS binder. The binder creates a load module if the output library is a partitioned data set (PDS). It creates a program object if the output library is a PDSE.

**CPP LE PRELINKER PHASE field**
The value of the CPP LE PRELINKER PHASE field indicates whether to customize the DSNHCPP and DSNHCPP2 language procedure to include a procedure step that executes the IBM Language Environment prelinker.
Acceptable values: NO or YES
Default: NO
Update: cannot change during update
DSNZPxxx: none

**NO**
Indicates that DSNHCPP and DSNHCPP2 are customized to prepare C++ programs without using the prelinker. Object code is passed directly from the compiler to the z/OS binder to create a program object, which must reside in a partitioned data set extended (PDSE). NO is the default setting.

**YES**
Indicates that DSNHCPP and DSNHCPP2 are customized to prepare C++ programs using the prelinker. Object code is passed from the compiler to the IBM Language Environment Prelinker and then to the z/OS binder. The binder creates a load module if the output library is a partitioned data set (PDS). It creates a program object if the output library is a PDSE.

**CPP CLASS LIB HEADERS field**
The value of the CPP CLASS LIB HEADERS field specifies the data set name for the C++ class header files.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: CBC.SCLBH.H
Update: cannot change during update
DSNZPxxx: none

Accept the default value if C++ is not available on your system.
- For C/C++ for z/OS, the data set name typically includes the qualifier CBC.SCLBH.H
- For C/C++ for MVS/ESA V3R2, the data set name typically includes the qualifier SCLB3H.HPP.

If you specify a name in this field, it is used to customize the DSNHCPP and DSNHCPP2 language procedures and the DSNH CLIST.

**CPP AUTO CALL LIBRARY field**
The value of the CPP AUTO CALL LIBRARY field specifies the data set name of the C++ auto call library.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: CEE.SCEECPP
Update: cannot change during update
DSNZPxxx: none

This data set name typically includes the qualifier SCEECPP. Accept the default value if C++ is not available on your system. The value of the CPP AUTO CALL LIBRARY field is used to customize the DSNHCPP and DSNHCPP2 language procedures and the DSNH CLIST.

**CPP CLASS LIBRARY field**
The value of the CPP CLASS LIBRARY field specifies the data set name of the C++ class library.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: CEE.SCLBCPP
Update: cannot change during update
DSNZPxxx: none

Accept the default value if C++ is not available on your system.
- For C/C++ for z/OS, the data set name typically includes the qualifier SCLBCPP.
- For C/C++ for MVS/ESA V3R2, the data set name typically includes the qualifier SCLB3CPP.

The value of the CPP CLASS LIBRARY field is used to customize the DSNHCPP and DSNHCPP2 language procedures and the DSNH CLIST.

**CPP PROCEDURE LIBRARY field**
The value of the CPP PROCEDURE LIBRARY field specifies the data set name for the C++ procedure library.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CEE.SCCNUTL
Update: cannot change during update
DSNZPxxx: none

Accept the default value if C++ is not available on your system.
- For C/C++ for z/OS, the data set name typically includes the qualifier SCCNUTL.
- For C/C++ for MVS/ESA V3R2, the data set name typically includes the qualifier SCBC3UTL.

**DSNTIPU1: Host language data set names panel 2**
The entries on panels DSNTIPU, DSNTIPU1, and DSNTIPW establish data set names for other product libraries. The values that you enter on these panels are edited into sample and installation jobs. Accept the default values for products that you do not have.

Data set names for other products appear in the jobs. These names are shown in the table in [Host language data set names panel 1: DSNTIPU](#) as they appear in the jobs that are shipped with Db2. Change the names of the data sets if they are different at your site.
<table>
<thead>
<tr>
<th>COBOL COMPILER LIBRARY field</th>
</tr>
</thead>
<tbody>
<tr>
<td>The value of the COBOL COMPILER LIBRARY field identifies the name of the compiler library that is to be used for COBOL.</td>
</tr>
<tr>
<td>Acceptable values:</td>
</tr>
<tr>
<td>Default:</td>
</tr>
<tr>
<td>Update:</td>
</tr>
<tr>
<td>DSNZP:</td>
</tr>
</tbody>
</table>

The data set name typically includes the qualifier SIGYCOMP. The value that you enter here is used to customize the DSNHICOB language procedure and the DSNH CLIST. You can leave this field blank if the compiler library is in the link list.

<table>
<thead>
<tr>
<th>COBOL LE PRELINKER PHASE field</th>
</tr>
</thead>
<tbody>
<tr>
<td>The value of the COBOL LE PRELINKER PHASE field indicates whether to customize the DSNHICOB language procedure to include a procedure step that executes the IBM Language Environment prelinker.</td>
</tr>
<tr>
<td>Acceptable values:</td>
</tr>
<tr>
<td>Default:</td>
</tr>
<tr>
<td>Update:</td>
</tr>
<tr>
<td>DSNZP:</td>
</tr>
</tbody>
</table>

**NO**
- Specifies that the DSNHICOB procedure is customized to prepare COBOL programs without using the prelinker. NO is required when using IBM Enterprise COBOL Version 5.1 and later releases. It is acceptable for older versions of the COBOL compiler that are supported by this release of Db2. Additional DD statements are allocated for the COBOL compiler and object code is passed directly from the compiler to the z/OS binder to create a program object, which must reside in a partitioned data set extended (PDSE). NO is the default setting.

**YES**
- Specifies that the DSNHICOB procedure is customized to prepare COBOL programs using the prelinker. YES is acceptable only when using IBM Enterprise COBOL Version 4.2 and older releases. Object code is passed from the compiler to the IBM Language Environment prelinker and then to the
z/OS binder. The binder creates a load module if the output library is a partitioned data set (PDS). It creates a program object if the output library is a PDSE.

FORTRAN COMPILER LIBRARY field
The FORTRAN COMPILER LIBRARY field identifies the data set name of the Fortran compiler library that is to be used by the Db2 installation process.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: none
Update: cannot change during update
DSNZPxxx: none

If you specify a value here, it is used to customize the DSNHFOR language procedure and the DSNH CLIST. You can leave this field blank if the compiler library is in the link list.

FORTRAN LINK EDIT LIB field
The FORTRAN LINK EDIT LIB field identifies the data set name of the Fortran linkage editor library that is to be used by the Db2 installation process.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: SYS1.VSF2FOR
Update: cannot change during update
DSNZPxxx: none

The value that you enter here is used to customize the DSNHFOR language procedure and the DSNH CLIST. Accept the default value if FORTRAN is not available on your system, and do not run IVP job DSNTIJ2F.

PL/I COMPILER LIBRARY field
The value of the PL/I COMPILER LIBRARY field specifies the name of the PL/I compiler library that is to be used by installation and sample jobs.

Acceptable values: blank, or valid data set name: see Valid data set name
Default: none
Update: cannot change during update
DSNZPxxx: none

For Enterprise PL/I, the data set name typically includes the qualifier SIBMZCMP. The value that you enter here is used to customize the DSNHPLI language procedure and the DSNH CLIST. You can leave the field blank if the PL/I compiler library is in the system link list.

PL/I LE PRELINKER PHASE field
The value of the PL/I LE PRELINKER PHASE field indicates whether to customize the DSNHPLI language procedure to include a procedure step that executes the IBM Language Environment prelinker.

Acceptable values: NO or YES
Default: NO
Update: cannot change during update
DSNZPxxx: none
Indicates that DSNHPLI is customized to prepare PL/I programs without using the prelinker. Object code is passed directly from the compiler to the z/OS binder to create a program object, which must reside in a partitioned data set extended (PDSE).

Indicates that DSNHPLI is customized to prepare PL/I programs using the prelinker. Object code is passed from the compiler to the IBM Language Environment prelinker and then to the z/OS binder. The binder creates a load module if the output library is a partitioned data set (PDS). It creates a program object if the output library is a PDSE.

DSNTIPW: Other data set names panel

The entries on the DSNTIPW panel establish data set names for the libraries of other products in your system. The values that are entered on this panel are edited into all pertinent sample and installation jobs.

If you do not have the product, accept the default. The default cannot be blank.

Many data set names for other products appear in the jobs. You can enter most of these data set names on this panel and on installation panel DSNTIPU and DSNTIPU1. These names are shown in the table in [Host language data set names panel: DSNTIPU] as they appear in the jobs that are shipped with . Change the names of the data sets if they are different at your site.

Press: ENTER to continue RETURN to exit HELP for more information

Figure 16. Other data set names panel: DSNTIPW

SYSTEM MACLIB field

The value of the SYSTEM MACLIB field specifies the data set name of the assembler macro library that is to be used by sample and installation jobs.

Acceptable values: valid data set name: see Valid data set name
Default: SYS1.MACLIB
Update: cannot change during update
DSNZPxx: none
**SYSTEM PROCEDURES field**
The value of the SYSTEM PROCEDURES field specifies the data set name of the system procedures library that is to be used by sample and installation jobs.

Acceptable values: valid data set name: see [Valid data set name]
Default: SYS1.PROCLIB
Update: cannot change during update
DSNZPxxx: none

**SORT LIBRARY field**
The value of the SORT LIBRARY field specifies the name of the data set where the DFSORT load module resides.

Acceptable values: valid data set name: see [Valid data set name]
Default: SYS1.SORTLIB
Update: cannot change during update
DSNZPxxx: none

If you use the Db2 Sort for z/OS product, Db2 does not use the value of this field. However, you should set this value to a valid data set name in case you need to use DFSORT as a fallback. The following guidance applies if Db2 Sort for z/OS is not installed.

If you use DFSORT, and your load library is not in the link list, you can change the DSNUPROC JCL procedure in job DSNTIJMV.

If DFSORT is installed as your primary z/OS sort product, you do not need to take any action to make DFSORT available to Db2. If multiple releases of DFSORT are installed, ensure that DFSORT R14 is found first in the system search order.

If DFSORT is not installed as your primary z/OS sort product, use one of the following methods to enable Db2 to use DFSORT:

- Add the DFSORT SORTLPA library to the link pack area list; then add the DFSORT SICELINK library to the link list.
- Add the DFSORT SICELINK library to the link list; then add the DFSORT SORTLPA library to the link list.

**Important:** If any non-IBM primary sort product is installed in the link list, install DFSORT in the link list, in system search order, after the non-IBM primary sort product.

**Recommendation:** If any non-IBM primary sort product is installed in the link pack area, add the DFSORT libraries to the link list.

- Add the DFSORT SICELINK library to the JOBLIB statement; then add the DFSORT SORTLPA library to the JOBLIB statement.
- Add the DFSORT SICELINK library to the STEPLIB DD statement; then add the DFSORT SORTLPA library to the STEPLIB DD statement.
- Add the DFSORT modules to a private library that is equivalent to one of the above configurations.

**Important:** If your non-IBM primary sort product is run from a private library, you must use DFSORT in the same way.
If you install DFSORT in the link pack area or link library, you must install it after you install the non-IBM primary sort product.

Db2 uses only the SORT and MERGE functions in DFSORT. If you want to use DFSORT for any other uses outside of this limited Db2 support, you must separately order and license DFSORT.

Related information:

[DFSORT Installation and Customization]

**IMS RESLIB field**
The IMS RESLIB field identifies the data set name of the IMS linkage editor library that is to be used by sample and installation jobs.

| Acceptable values: | valid data set name: see [Valid data set name] |
| Default:           | none |
| Update:            | cannot change during update |
| DSNZPxxx:          | none |

If you do not have IMS, you do not need to connect Db2 to IMS, and you can skip the phase 4 sample application jobs DSNTEJ4C and DSNTEJ4P.

**ISPF ISPLINK MODULE field**
The ISPF ISPLINK MODULE field identifies the data set name of the ISPF load module library that is to be used by sample and installation jobs.

| Acceptable values: | valid data set name: see [Valid data set name] |
| Default:           | ISP.SISPLOAD |
| Update:            | cannot change during update |
| DSNZPxxx:          | none |

**GDDM MACLIB field**
The GDDM MACLIB field identifies the data set name of the GDDM macro library that is to be used by sample and installation jobs.

| Acceptable values: | valid data set name: see [Valid data set name] |
| Default:           | GDDM.SADMSAM |
| Update:            | cannot change during update |
| DSNZPxxx:          | none |

This field and the GDDM LOAD MODULES field must both have a valid data set name or must both be blank. The data set name that you specify in this field is included in the compiler step SYSLIB DD concatenations of DSNHASM, DSNHC, DSNHCOB, DSNHCOB2, DSNHICOB, DSNHICB, and DSNHPLI. The installation CLIST only generates sample jobs DSNTEJ75 and DSNTEJ78 if you specify a GDDM MACLIB name.

**GDDM LOAD MODULES field**
The GDDM LOAD MODULES field identifies the data set name of the GDDM load module library that is to be used by sample and installation jobs.

| Acceptable values: | valid data set name: see [Valid data set name] |
| Default:           | GDDM.SADMMOD |
| Update:            | cannot change during update |
| DSNZPxxx:          | none |
This field and the GDDM MACLIB field must both have a valid data set name or must both be blank. The data set name that you specify in this field is included in the link-edit SYSLIB concatenations of DSNHASM, DSNHC, DSNHCOB, DSNHCOB2, DSNHICOB, DSNHICB2, and DSNHPLI.

**CICS LOAD LIBRARY field**

The CICS LOAD LIBRARY field identifies the data set name that is to be used for the CICS load module library.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CICSTS.SDFHLOAD
Update: cannot change during update
DSNZP: none

If you do not use CICS, use a blank in the CICS LOAD LIBRARY field.

**CICS MACRO LIBRARY field**

The CICS MACRO LIBRARY field identifies the data set name that is to be used for the CICS macro library.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CICSTS.SDFHMAC
Update: cannot change during update
DSNZP: none

**CICS COBOL LIBRARY field**

The CICS COBOL LIBRARY FIELD specifies the data set name for the CICS library that the COBOL programs are to use.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CICSTS.SDFHCOB
Update: cannot change during update
DSNZP: none

**CICS PL/I LIBRARY field**

The CICS PL/I LIBRARY field identifies the data set name for the CICS library that PL/I programs are to use.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CICSTS.SDFHPLI
Update: cannot change during update
DSNZP: none

**CICS EXCI LIBRARY field**

The CICS EXCI LIBRARY field identifies the data set name for the CICS library that contains the CICS EXCI load modules.

Acceptable values: valid data set name: see [Valid data set name](#)
Default: CICSTS.SDFHEXCI
Update: cannot change during update
DSNZP: none
DSNTIPD: Sizes panel 1

The entries on the DSNTIPD panel establish the size of the Db2 catalog, directory, and log data sets. Entries on the DSNTIPD panel also define the amount of storage that can be used for storing large object (LOB) values.

The values that you supply on this panel are estimates that are used in calculating sizes for main storage and data sets. The values do not reduce any system limits and do not preclude an application or user from exceeding these estimates, within reasonable limits. For example, if you specify 500 databases, you could create 600. However, if you exceed the values by a large margin, you might encounter a shortage of main storage or use many secondary extents for some data sets. You can usually change the main storage values by using the next panel. If you cannot change these values, see Updating subsystem parameter and application default values for the appropriate method.

The installation CLIST contains formulas that calculate the space for each catalog data set and the indexes that Db2 requires for each data set. Data that you enter on this panel is used in these formulas. Use integers. Do not enter fractions. You can use K (as in 32 K) for multiples of 1024 bytes and M (as in 16 M) for multiples of 1,048,576 bytes in most fields, but do not exceed the maximum value that is accepted by the field. For example, for the PACKAGE LISTS field, which has a maximum of 32,000, you can enter 31 K, meaning 31,744 bytes. Values of 32 K and above exceed the maximum acceptable value for this field.

Many of the fields on this panel affect the values of the EDMSTATC and EDMDBDC parameters in macro DSN6SPRM.

If you are migrating, Db2 12 adopts the Db2 catalog, directory, BSDS, and active logs from your previous version. Therefore, during migration, you cannot change any of the fields on this panel that affect those data sets.

Updating the parameters:

You can alter the characteristics of the Db2 catalog, directory, BSDS, and active and archive logs by using the methods described on Updating subsystem parameters that cannot be updated by using DSNTIPB. You cannot actually change the values of these parameters.
Related tasks:
- Calculating storage requirements with the installation CLIST

Related reference:
- DSNTIPC: CLIST calculations panel 1

**DATABASES field for panel DSNTIPD**

The value of the DATABASES field of panel DSNTIPD indicates the estimated number of user databases that are to exist in your subsystem.

- **Acceptable values:** 1 - 64000
- **Default:** 200
- **Update:** see Updating subsystem parameter and application default values; cannot update during migration
- **DSNZPxxx:** none

**TABLES field**

The value of the TABLES field indicates the estimated average number of tables that is to exist for each database in your subsystem.

- **Acceptable values:** 1 - 400
- **Default:** 20
- **Update:** see Updating subsystem parameter and application default values; cannot update during migration
- **DSNZPxxx:** none

**COLUMNS field**

The value of the COLUMNS field identifies the average number of columns per table in your subsystem. This value is to be used in calculating sizes for main storage and data sets.

- **Acceptable values:** 1 - 750
- **Default:** 10
- **Update:** see Updating subsystem parameter and application default values; cannot update during migration
**VIEWS field**
The value of the VIEWS field specifies the estimated average number of views that are to exist for each table in your subsystem.

Acceptable values: 1 - 200
Default: 3
Update: see Updating subsystem parameter and application default values

**TABLE SPACES field**
The value of the TABLE SPACES field indicates the estimated average number of table spaces that is to exist per database in your subsystem.

Acceptable values: 1 - 400
Default: 20
Update: see Updating subsystem parameter and application default values; cannot update during migration

**PLANS field**
The value of the PLANS field is the estimated number of application plans that are to exist in your subsystem. Each program requires a separate application plan.

Acceptable values: 1 - 32000
Default: 200
Update: see Updating subsystem parameter and application default values; cannot update during migration

**PLAN STATEMENTS field**
The value of the PLAN STATEMENTS field specifies the estimated average number of SQL statements that are to exist for each application plan.

Acceptable values: 1 - 32000
Default: 30
Update: see Updating subsystem parameter and application default values; cannot update during migration

**PACKAGES field**
The value of the PACKAGES field specifies the estimated number of packages that are to exist in the system.

Acceptable values: 1 - 256000
Default: 300
Update: see Updating subsystem parameter and application default values; cannot update during migration
PACKAGE STATEMENTS field
The value of the PACKAGE STATEMENTS field specifies the estimated number of individual SQL statements that are to exist for each package.

Acceptable values: 1 - 32000
Default: 10
Update: see Updating subsystem parameter and application default values; cannot update during migration
DSNZPxxx: none

PACKAGE LISTS field
The value of the PACKAGE LISTS field is the average number of packages that are estimated to be in a package list for each plan.

Acceptable values: 1 - 32000
Default: 2
Update: see Updating subsystem parameter and application default values; cannot update during migration
DSNZPxxx: none

EXECUTED STMTS field
The value of the EXECUTED STMTS field is an estimate of the average number of SQL statements that are to be executed per plan.

Acceptable values: 1 - 32000
Default: 15
Update: see Updating subsystem parameter and application default values; not during migration
DSNZPxxx: none

The number of SQL statements that are executed can be less than the number of statements that are written.

TABLES IN STMT field
The value of the TABLES IN STMT field indicates the estimated average number of tables that is to be used per SQL statement.

Acceptable values: 1 - 16
Default: 2
Update: see Updating subsystem parameter and application default values; cannot update during migration
DSNZPxxx: none

Some SQL statements use more than one table (for example, those using joins, unions, or subselect clauses). Consider how often you expect to use such statements when choosing a value for this parameter.

LOB INLINE LENGTH field (LOB_INLINE_LENGTH subsystem parameter)
The LOB_INLINE_LENGTH subsystem parameter specifies the default length (in bytes) that is to be used for inline LOB columns.

An inline LOB is the portion of a LOB column that resides in the containing table's base table space rather than in the column's auxiliary table space. The value in this
field is used when the INLINE LENGTH option is not specified in the definition of a LOB column in a CREATE TABLE or ALTER TABLE ADD statement.

Acceptable values: 0 - 32680
Default: 0
Update: option 13 on panel DSNTPB
DSNZPxxx: DSN6SYSP LOB_INLINE_LENGTH

When you enter a value in this field, keep the following points in mind:

- A value of 0 causes LOB columns to be created with no inline attribute by default.
- Odd values are rounded to the next even integer for any DBCLOB column created. For example, if you enter 1023, then 1024 bytes of data for new DBCLOB columns reside in the base table.
- The value of this parameter is in bytes, regardless of the LOB column’s data type or subtype.

Related tasks:
- Improving performance for LOB data (Db2 Performance)

Related reference:
- CREATE TABLE (Db2 SQL)
- ALTER TABLE (Db2 SQL)

DSNTIP7: SQL OBJECT DEFAULTS PANEL 1

The entries on the DSNTIP7 panel provide general default settings for creating table spaces.

Figure 18. SQL OBJECT DEFAULTS PANEL 1: DSNTIP7
OBJECT CREATE FORMAT field (OBJECT_CREATE_FORMAT subsystem parameter)
The OBJECT_CREATE_FORMAT subsystem parameter specifies whether Db2 is to create new table spaces and indexes to use a basic or extended log record format.

Acceptable values: BASIC, EXTENDED
Default: EXTENDED
Update: option 14 on panel DSNTIP7
DSNZPxxx: DSN6SPRM.OBJECT_CREATE_FORMAT

BASIC
New table spaces and indexes are created with a maximum of 256 TB of log record addressing capacity over the life of a Db2 subsystem, or a maximum log record sequence number (LRSN) of 2**48 over the life of a Db2 data-sharing group.

EXTENDED
New table spaces and indexes are created with a maximum of 1 YB of log record addressing capacity over the life of a Db2 subsystem or a maximum LRSN of 2**80 over the life of the Db2 data-sharing group. This setting is the default and is recommended in either of the following situations:

- When Db2 is likely to exhaust the basic log format. This setting is required to update database objects if the Db2 log RBA exceeds x'FFFFFFFFFFFFFFF' for non-data-sharing environments or if the LRSN exceeds 2**48 for data-sharing environments.
- In data-sharing environments only, when duplicate LRSN values occur because the processing speed exceeds the precision of the traditional log addressing format.

Subsystem parameter OBJECT_CREATE_FORMAT does not affect new partitions that are added to existing partition-by-growth table spaces. New partitions use the RBA format of the previous logical partition.

Related tasks:
- Converting page sets to the 10-byte RBA or LRSN format (Db2 Administration Guide)

Related reference:
- UTILITY OBJECT CONVERSION field (UTILITY_OBJECT_CONVERSION subsystem parameter)

UTILITY OBJECT CONVERSION field (UTILITY_OBJECT_CONVERSION subsystem parameter)
The value of the UTILITY_OBJECT_CONVERSION parameter specifies whether Db2 utilities that accept the RBALRSN_CONVERSION option will convert existing table spaces and indexes to 6-byte page format, to a 10-byte page format, or perform no conversion.

The default behavior normally applies when the utility control statement does not specify the RBALRSN_CONVERSION option. Alternatively, the UTILITY_OBJECT_CONVERSION parameter can also be used to prevent use of RBALRSN_CONVERSION options to convert existing table spaces and indexes to 6-byte page format.

Acceptable values: BASIC, EXTENDED, or NOBASIC, NONE
Default: NONE
Update: option 14 on panel DSNTIP7
DSNZPxxx: DSN6SPRM.UTILITY_OBJECT_CONVERSION

BASIC

Existing table spaces and indexes that use extended 10-byte page format are converted to basic 6-byte page format. BASIC is allowed only if the OBJECT CREATE FORMAT field is also set to BASIC.

EXTENDED

Existing table spaces and indexes that use 6-byte page format are converted to extended 10-byte page format. EXTENDED is allowed only if the OBJECT CREATE FORMAT field is also set to EXTENDED.

NOBASIC

Existing table spaces and indexes that use 6-byte page format are converted to extended 10-byte page format. Table spaces and indexes that already use extended 10-byte page format cannot be returned to the 6-byte page format. When this setting is in effect, utilities that specify the RBALRSN_CONVERSION keyword with BASIC fail. In addition, utilities that specify RBALRSN_CONVERSION keyword with NONE when the object is in 6-byte page format fail. NOBASIC is allowed in this field only if the OBJECT CREATE FORMAT field is set to EXTENDED. If a value is not specified for RBALRSN_CONVERSION, the RBALRSN_CONVERSION value defaults to EXTENDED.

NONE

No conversion is performed. This option is the default setting of this parameter. NONE is allowed regardless of the OBJECT CREATE FORMAT setting.

VARY DS CONTROL INTERVAL field (DSVCI subsystem parameter)

The DSVCI subsystem parameter controls whether Db2-managed data sets that are created by CREATE TABLESPACE or CREATE INDEX statements are to have variable VSAM control intervals.

Acceptable values: YES, NO
Default: YES
Update: option 14 on panel DSNTIPB
DSNZPxxx: DSN6SYSP DSVCI

YES

Db2 creates Db2-managed data sets for table spaces and index spaces. The Db2-managed data sets have a VSAM control interval that corresponds to the buffer pool that is used for the table space or index space.

NO

Db2-managed data sets are created with a fixed control interval of 4 KB, regardless of the buffer pool size.

This parameter is online-updatable. If you change this value from NO to YES, any pre-existing or migrated data sets remain in 4 KB control intervals until they are redefined. If you change this value from YES to NO, any pre-existing or migrated data sets in 8 KB, 16 KB, or 32 KB control intervals remain in those control intervals until they are redefined. You can explicitly redefine a data set. In addition, data sets are implicitly redefined by utilities such as LOAD REPLACE, REORG TABLESPACE, or RECOVER.
TABLE SPACE ALLOCATION field (TSQTY subsystem parameter)
The TSQTY subsystem parameter specifies the amount of space in KB that is to be allocated for primary and secondary spaces. This parameter applies to Db2-defined data sets for table spaces that are created without the PRIQTY and SECQTY clauses.

Acceptable values: 0 - 4194304
Default: 0
Update: option 14 on panel DSNTIPB
DSNZP: DSN6SYSP TSQTY

A value of 0 indicates that Db2 is to use a default value of 1 cylinder for a non-LOB table space or 10 cylinders for a LOB table space.

In a data sharing environment, this parameter has group scope.

Related concepts:
[How SQL statements are invoked (Db2 SQL)]

INDEX SPACE ALLOCATION field (IXQTY subsystem parameter)
The IXQTY subsystem parameter controls the amount of space in KB that is to be allocated for primary and secondary spaces. This parameter applies to Db2-defined data sets for index spaces that are created without the PRIQTY and SECQTY clauses.

Acceptable values: 0 - 4194304
Default: 0
Update: option 14 on panel DSNTIPB
DSNZP: DSN6SYSP IXQTY

A value of 0 indicates that Db2 is to use a default allocation of one cylinder.

In a data sharing environment, this parameter has group scope.

OPTIMIZE EXTENT SIZING field (MGEXTSZ subsystem parameter)
The MGEXTSZ subsystem parameter controls whether secondary extent allocations for Db2-managed data sets are to be sized according to a sliding scale. Sizing the secondary extents on a sliding scale optimizes the likelihood of reaching the maximum data set size before secondary extents are exhausted.

Acceptable values: YES, NO
Default: YES
Update: option 14 on panel DSNTIPB
DSNZP: DSN6SYSP MGEXTSZ

YES
Db2 automatically optimizes the secondary extent allocations of data sets for table spaces and index spaces that have a SECQTY value of greater than zero. When all secondary extents are exhausted for the first data set of a nonpartitioned table space or a nonpartitioned index space that has a SECQTY value of greater than zero, the primary space allocation of each subsequent data set is the larger of the SECQTY setting and the value that is derived from the sliding scale algorithm.
NO Secondary extent allocations are managed manually. For nonpartitioned table
spaces and nonpartitioned index spaces, when all extents of the first data set
are exhausted, the primary space allocation of each subsequent data set is
always the PRIQTY setting.

When the sliding scale is used, secondary extent allocations that are allocated
earlier are smaller than those allocated later, until a maximum allocation is
reached. The maximum allocation is 127 cylinders for data sets with a maximum
size of 16 GB or less, and 559 cylinders for data sets with a maximum size of 32
GB or 64 GB.

Secondary extent allocations for data sets in implicitly created table spaces are not
influenced by this setting because they always use the sliding scale.

In migration mode, the input migration value for this field is overridden to YES,
so you need to reset it to NO if you want to continue to disable secondary extent
optimization.

**TS_COMPRESSION_TYPE field (TS_COMPRESSION_TYPE subsystem parameter)**
The TS_COMPRESSION_TYPE subsystem parameter specifies the type of
compression algorithm to use when Db2 creates new compressed table spaces,
loads into, and reorganizes existing compressed table spaces.

Acceptable values: FIXED_LENGTH, HUFFMAN

Default: FIXED_LENGTH

Update: option 14 on panel DSNTIPB

DSNZPxxx: DSN6SPRM.TS_COMPRESS_TYPE

**FIXED_LENGTH**
Db2 uses its existing compression algorithm to perform compression and
decompression of data stored in the table space. This is the default setting.

**HUFFMAN**
Db2 uses Huffman compression provided by IBM z14™ or later hardware to
perform compression and decompression of data stored in the table space.
Huffman compression only applies to universal table spaces. All other table
space types will use fixed-length compression regardless of the
TS_COMPRESSION_TYPE setting.

**DSNTIP71: SQL OBJECT DEFAULTS PANEL 2**
The first entry on the DSNTIP71 panel provides a general default setting for
creating table spaces. The remaining fields define default values for the
system-required objects that Db2 creates automatically. Db2 uses these options for
any indexes or table spaces that it implicitly creates.
The DPSEGSZ subsystem parameter specifies the default segment size in pages for a CREATE TABLESPACE statement that omits the SEGSIZE clause. Acceptable values: 0, 4 - 64 (multiples of 4) Default: 32 Update: option 15 on panel DSNTIPB DSNZPxxx: DSN6SYSP.DPSEGSZ

The DPSEGSZ subsystem parameter has no effect for the creation of table spaces in a work file database or implicitly created table spaces. For explicit CREATE TABLESPACE statements that omit the SEGSIZE clause, it has the following effects:

0 The default value is SEGSIZE 32.
4 - 64 (multiples of 4) The default SEGSIZE value is the specified value.

Related concepts:
- Table space types and characteristics in Db2 for z/OS (Db2 Administration Guide)

Related reference:
- CREATE TABLESPACE (Db2 SQL)

Related information:
- Implementing Db2 table spaces (Db2 Administration Guide)

The PCTFREE_UPD subsystem parameter specifies the default value to use for the PCTFREE FOR UPDATE clause of CREATE TABLESPACE or ALTER TABLESPACE statements. It specifies the default amount of free space to reserve on each page for use by subsequent UPDATE operations when data is added to the table by INSERT operations or utilities.
Acceptable values: AUTO, 0 - 99
Default: 0
Update: option 15 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.PCTFREE_UPD
Data sharing scope: The recommendation is to use the same value on all members.

This value is used as the default FOR UPDATE value when no PCTFREE FOR UPDATE clause is specified for a CREATE TABLESPACE or ALTER TABLESPACE statement.

**AUTO**
Db2 uses real-time statistics values to automatically calculate the percentage of free space that is to be used by update operations. This value is equivalent to specifying PCTFREE FOR UPDATE -1 in the CREATE TABLESPACE or ALTER TABLESPACE statement.

**0 - 99**
Db2 reserves the specified percentage of space for use by update operations.

Related tasks:
- Reserving free space for table spaces (Db2 Performance)

Related reference:
- [CREATE TABLESPACE (Db2 SQL)]
- [ALTER TABLESPACE (Db2 SQL)]

**PAGE SET PAGE NUMBERING field (PAGESET_PAGENUM subsystem parameter)**
The PAGESET_PAGENUM subsystem parameter specifies whether partition-by-range table spaces and associated partitioned indexes are created to use absolute page numbers across partitions or relative page numbers.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: ABSOLUTE or RELATIVE
Default: ABSOLUTE
Update: option 15 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.PAGESET_PAGENUM
Data sharing scope: Use the same value on all members.

**ABSOLUTE**
Partition-by-range table spaces and associated indexes are created with absolute page numbers.

**RELATIVE**
Partition-by-range table spaces are created so that PGNUM in the page header has no partition number, and the partition number is contained only in the header page for the partition. The same numbering system applies to partitioned indexes on a table in a table space with relative page numbering.

Related reference:
- [CREATE TABLESPACE (Db2 SQL)]
- [CREATE TABLE (Db2 SQL)]
RETRY STOPPED OBJECTS field (RETRY_STOPPED_OBJECTS subsystem parameter)
The RETRY_STOPPED_OBJECTS subsystem parameter specifies whether Db2 immediately rejects requests for a stopped object or rechecks whether the object is started, up to the IRLM timeout limit.

- **Acceptable values:** NO, YES
- **Default:** NO
- **Data sharing scope:** It is recommended but not required that all members use the same setting.
- **Online changeable:** Yes
- **Update:** Option 15 on panel DSNTPB
- **DSNZPxxx:** DSN6SPRM.RETRY_STOPPED_OBJECTS

**NO** Db2 immediately rejects requests for stopped objects.

**YES** Db2 retries requests for stopped objects in case the object is restarted, up to IRLM timeout limit, which is specified by the value of the IRLMRWT subsystem parameter.

**Related reference:**
RESOURCE TIMEOUT field (IRLMRWT subsystem parameter)

PAD INDEXES BY DEFAULT field (PADIX subsystem parameter)
The PADIX subsystem parameter determines whether new indexes are to be padded by default.

- **Acceptable values:** NO, YES
- **Default:** NO
- **Update:** option 15 on panel DSNTPB
- **DSNZPxxx:** DSN6SPRM.PADIX

**NO** New indexes will not be padded unless the PADDED option is specified on the CREATE INDEX statement.

This is the default value.

**YES** New indexes will be padded unless the NOT PADDED option is specified on the CREATE INDEX statement.

This parameter affects only indexes that have at least one varying-length column.

**Related concepts:**
Indexes that are padded or not padded (Introduction to Db2 for z/OS)

INDEX MEMORY CONTROL field (INDEX_MEMORY_CONTROL subsystem parameter)
The INDEX_MEMORY_CONTROL subsystem parameter specifies the amount of memory that Db2 allocates for fast traversing of indexes (sometimes called “fast traverse blocks” or “FTB”).

Db2 12 introduces this subsystem parameter. It takes effect in function level 100. However, during coexistence with Db2 11, no storage is allocated for fast index traversal of group buffer pool dependent indexes.

- **Acceptable values:** AUTO, DISABLE, or 10 - 200000
Db2 sets the upper limit to 20 percent of the currently allocated buffer pool storage, or 10 MB, whichever is larger.

DISABLE

Db2 returns any existing storage that is allocated for fast index traversal and allocates no more storage for fast index traversal.

10 - 200000

Specifies the storage limit for fast index traversal, in megabytes.

Related concepts:

- Fast index traversal (Db2 Performance)

Related tasks:

- Enabling or disabling fast index traversal at the index level (Db2 Performance)

**DDL MATERIALIZATION field (DDL_MATERIALIZATION subsystem parameter)**

The DDL_MATERIALIZATION subsystem parameter specifies when Db2 materializes changes to object definitions.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

It applies only for ALTER TABLE statements that specify the ALTER COLUMN and SET DATA TYPE keywords to change the data type, length, precision, or scale of a column. The column must be in a table in a universal table space, or in a table space that has an unmaterialized pending alter to convert it to a universal table space. The data sets for the table space must already be created.

This option is ignored in other cases, such as when alter operations must be executed as a pending alteration or must be executed as an immediate alteration.

Acceptable values: ALWAYS_IMMEDIATE, ALWAYS_PENDING

Default: ALWAYS_IMMEDIATE

Data sharing scope: It is recommended that all members use the same setting

Update: Option 15 on panel DSNTIPB

DSNZPxxx: DSN6SPRM.DDL_MATERIALIZATION

This option specifies whether object definition changes are executed as immediate or pending alterations:

1 (ALWAYS_IMMEDIATE)

For applicable requests, changes are materialized at the time that the request is executed, and the containing table space is placed in AREO* or REORP status.

If there are any existing unmaterialized pending changes, the request fails.

2 (ALWAYS_PENDING)
For applicable requests, changes are not materialized at the time that the
request is executed, and the affected objects are available until it is convenient
to implement the changes. The containing table space is placed in AREOR
status. If any immediate options are specified in the same statement, the
change request fails. If any subsequent immediate changes are executed before
a pending change is materialized, those subsequent immediate changes fail.

Related concepts:

- Pending data definition changes (Db2 Administration Guide)

Related reference:

- ALTER TABLE (Db2 SQL)

**DEFAULT INSERT ALGORITHM field**

**DEFAULT_INSERT_ALGORITHM subsystem parameter**

The DEFAULT_INSERT_ALGORITHM subsystem parameter specifies the default
algorithm for inserting data into table spaces.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated
function level is function level 500 or higher.

Acceptable values: 0, 1, 2

Default: 2

Data sharing scope: It is recommended that all members use the same
setting.

Online changeable: Yes

DSNZPxxx: DSN6SPRM.DEFAULT_INSERT_ALGORITHM

The following list describes how the DEFAULT_INSERT_ALGORITHM subsystem
parameter setting is applied in relation to the INSERT ALGORITHM parameter of
the CREATE TABLESPACE and ALTER TABLESPACE statements:

0  The basic insert algorithm (insert algorithm 1) is always used regardless of any
    INSERT ALGORITHM parameter setting that might have been specified when
    the table space was created or altered.

1  The basic insert algorithm is used unless the INSERT ALGORITHM parameter
    was set to 2 when the table space was created or altered.

2  Insert algorithm 2 is used unless the INSERT ALGORITHM parameter was set
to 1 when the table space was created or altered.

Related concepts:

- Insert algorithm 2 (Db2 Performance)

Related reference:

- CREATE TABLESPACE (Db2 SQL)
- ALTER TABLESPACE (Db2 SQL)

**DSNTIP72: SQL OBJECT DEFAULTS PANEL 3**

The entries on the DSNTIP72 panel provide options for accessing stopped objects,
and options for renaming tables.
DEFINE DATA SETS field (IMPDSDEF subsystem parameter)

The IMPDSDEF subsystem parameter specifies whether Db2 is to define the underlying data set for certain implicitly created table spaces and indexes.

This parameter applies to:
- Base table spaces
- Index spaces of indexes that are implicitly created on base tables
- Implicitly created LOB or XML table spaces
- Auxiliary indexes
- Node ID indexes
- Document ID indexes

Acceptable values: YES, NO
Default: YES
Update: option16 on panel DSNTIPB
DSNZPxxx: DSN6SYSP IMPDSDEF

YES Underlying data sets are defined when the table space or index space is created. This is the default setting.

NO Underlying data sets are defined when data is first inserted into the table or index.

However, if all of the following conditions are met, underlying data sets are defined when the table space or index space is created:
- The table contains an LOB column.
- The table is in a clone relationship.
- The containing table space is a partition-by-growth table space.

USE DATA COMPRESSION field (IMPTSCMP subsystem parameter)

The IMPTSCMP subsystem parameter determines whether a table space that is created in an implicitly created database is to use data compression.
Acceptable values: NO, YES  
Default: NO  
Update: option 16 on panel DSNTIPB  
DSNZP: DSN6SYSP IMPTSCMP

NO  Disables data compression. This is the default value.  
YES  Enables data compression for table spaces that are created in implicitly created databases.

The value of the USE DATA COMPRESSION field applies only to implicitly created base table spaces. It is not used for implicitly created LOB or XML table spaces.

**PREVENT ALTER LIMITKEY field (PREVENT_ALVERTB_LIMITKEY subsystem parameter)**  
PREVENT_ALVERTB_LIMITKEY determines whether Db2 disallows altering the limit key by using an ALTER TABLE statement for index-controlled partitioned table spaces. This alter operation places the table space in REORG-pending (REORP) restrictive status, and the data is unavailable until the affected partitions are reorganized. Use PREVENT_ALVERTB_LIMITKEY to avoid this data unavailability.

Acceptable values: NO, YES  
Default: NO  
Update: option 16 on panel DSNTIPB  
DSNZP: DSN6SPRM PREVENT_ALVERTB_LIMITKEY

NO  Specifies that you can alter a limit key by using an ALTER TABLE statement for index-controlled partitioned table spaces.  
NO is the default.

YES  Specifies that altering a limit key by using an ALTER TABLE statement for index-controlled partitioned table spaces is not permitted. An ALTER TABLE statement must not attempt to alter the limit key for an index-controlled partitioned table.

Related tasks:
- [Creating tables partitioned by data value ranges (Db2 Administration Guide)](#)  
- [Changing the boundary between partitions (Db2 Administration Guide)](#)

Related reference:
- [REORG-pending status (Db2 Utilities)](#)

**EXCLUDE PART KEY COLUMNS field (IX_TB_PART_CONV_EXCLUDE subsystem parameter)**  
Specifies whether to exclude trailing columns from the table-controlled partitioning keys when table spaces are converted from index-controlled partitioning to table-controlled partitioning.

Acceptable values: YES, NO  
Default: YES  
Update: option 16 on panel DSNTIPB  
DSNZP: DSN6SPRM IX_TB_PART_CONV_EXCLUDE
YES

When converting from index-controlled partitioning to table-controlled partitioning, Db2 defines the new partitioning key to use only significant columns of the original partitioning index by default. If all partitions contain 'FF'x for a column in the internal limit key value that is stored in the table's SYSINDEXPART.LIMITKEY entry, Db2 excludes that column. Such columns were specified with a value of 'FF'x, MAXVALUE, MINVALUE, or no value when the index-controlled partitioning index was defined. However, if a new limit key value is specified for a column in an ALTER TABLE statement that is issued in one of the following cases, the column is used to define the new partitioning key regardless of whether the column was originally specified with a value of 'FF'x, MAXVALUE, or MINVALUE:

- To add a new partition
- To change a partition boundary
- To rotate a partition to last on an index-controlled partitioned table space

NO

When converting from index-controlled partitioning to table-controlled partitioning, Db2 uses all columns of the original partitioning index to define the new partitioning key. This definition includes the trailing columns of the original partitioning index that do not affect the partitioning.

This parameter does not affect the automatic conversion to table-controlled partitioning when a CREATE INDEX statement is issued with an ENDING AT clause and without a CLUSTER clause. In this case, the Db2 subsystem uses the partitioning key that is specified by the CREATE INDEX statement.

Learn more: For comprehensive background, how-to information, and examples for various paths for converting your deprecated “classic” partitioned (non-UTS) table spaces to partition-by-range table spaces, see the white paper Conversion from index-controlled partitioning to Universal Table Space (UTS).

Related tasks:

- Converting partitioned (non-UTS) table spaces to partition-by-range universal table spaces (Db2 Administration Guide)

Related reference:

- CREATE INDEX (Db2 SQL)

RENAME TABLE field (RENAMETABLE subsystem parameter)

The RENAMETABLE subsystem parameter specifies whether RENAME TABLE issues an error or allows the operation when RENAME TABLE is issued for a table that is referenced in a view definition or the definition of an SQL table function.

Acceptable values:
- DISALLOW_DEP_VIEW_SQLTUDF
- ALLOW_DEP_VIEW_SQLTUDF

Default: DISALLOW_DEP_VIEW_SQLTUDF

Update: option16 on panel DSNTIPB

DSNZPRM: DSN6SPRM RENAMETABLE

DISALLOW_DEP_VIEW_SQLTUDF

Db2 issues an error when RENAME TABLE is issued for a table that is referenced in the definition of a view or an SQL table function.

ALLOW_DEP_VIEW_SQLTUDF

Db2 does not issue an error when RENAME TABLE is issued for a table that is referenced in the definition of a view or an SQL table function.
When a RENAME TABLE statement is issued for such a table, the Db2 database manager takes the following actions:

- Updates internal information about object dependencies, but does not change the text representations of statements that created or altered views or routines that depend on the renamed table in the SYSIBM.SYSVIEWS or SYSIBM.SYSROUTINES catalog tables.
- Invalidates statements that refer to the table that is renamed.

In a data sharing environment, specify the same value of RENAMETABLE for all members.

Related reference:

- SYSROUTINES catalog table (Db2 SQL)
- SYSVIEWS catalog table (Db2 SQL)

DSNTIP9: Work file database panel

The fields on the DSNTIP9 panel configure the 4 KB and 32 KB page size table spaces in the Db2 work file database.

The values in these fields determine the following:

- The total amount of space that is available for each type of table space.
- The number of table spaces that are to be created during the installation or migration process for each type of table space. In INSTALL mode, you must specify at least 1 for each of the 4 types of work file table spaces: 4 KB sort work, 32 KB sort work, 4 KB temporary work, and 32 KB temporary work. In MIGRATE mode, you can specify 0 for any or all of these table spaces.
- The segment size of each type of table space.

The settings on this panel are used to tailor installation job DSNTIJTM, which creates the work file database. For more information about job DSNTIJTM, see “Installation step 18: Create default storage group, define temporary work files, and bind Db2 REXX Language Support: DSNTIJTM” on page 441. To determine the storage requirements for work file database table spaces, see Work file database storage requirements.

Sort work table space in the work file database are used as temporary space for SQL statements and triggers that require working storage. In particular, this includes statements that use the following types of operations:

- GROUP BY or HAVING (without index)
- ORDER BY (without index)
- DISTINCT (without index)
- UNION (except UNION ALL)
- EXISTS (subselect)
- ANY (subselect)
- SOME (subselect)
- ALL (subselect)
- Some joins

Temporary work table spaces in the work file database are used for declared global temporary tables. They can increase in size beyond the primary allocation. When
the value of the INSTALL TYPE field is INSTALL, temporary work table spaces are
created as either partition-by-growth or segmented table spaces, depending on the
amount of primary space per table space.

Unlike temporary work table spaces, sort work table spaces are always segmented
table spaces and cannot increase in size.

You can add a work file table space or change the size of an existing one by
deleting and redefining it. Db2 needs to be started for these activities and you do
not need to stop the work file database. All Db2 users share the work file database
table spaces. You cannot use utilities on the work file database table spaces.

You can create additional work file table spaces at any time, including during
migration. Creating additional work file table spaces can improve Db2 performance
by reducing device contention among applications that require working storage.

For temporary work only, the work file database can contain partition-by-growth

table spaces to increase the available space for declared global temporary tables

and reduce the possibility of receiving SQLCODE -904. A partition-by-growth table

space can grow up to 128 TB. The maximum number of partitions to which the

partition-by-growth table space can grow and the maximum size of the table space

are determined by the MAXPARTITIONS and DSSIZE values that are specified for

the table space.

**Important:** Db2 12 uses more 32KB work files than previous Db2 releases. For

migration to Db2 12, you might need to increase your 32KB work file allocations

for work file data sets and work file buffer pools.

**Recommendation:** If you have not already done so, drop work file table spaces for
temporary work and re-create them as partition-by-growth table spaces. (Existing
table spaces cannot be altered to become partition-by-growth table spaces.)

**Updating the parameters:** You can alter the characteristics of the Db2 catalog,
directory, work file databases, BSDS, and active and archive logs by using the
methods described on Updating subsystem parameter and application default values.
**Related tasks:**

- Creating additional work file table spaces to reduce contention (Db2 Performance)

**4K SORT PRIMARY SPACE field**

The value of the 4K SORT PRIMARY SPACE field specifies the total amount of space for all 4 KB sort work table spaces in the work file database. This value is specified in MB, but you can enter the value in GB by specifying a suffix of G, for example 1024G.

Unlike temporary work table spaces, sort work table spaces are always segmented table spaces and cannot increase in size.

Acceptable values: for installation: 1 - 32,702,464; for migration: 0 - 32,702,464

Default: for installation: 20; for migration: 0

Update: none

DSNZPxxx: none

The value of the 4K SORT PRIMARY SPACE field is divided by the value of the 4K SORT WORK TS’S field to determine the primary space quantity (PRIQTY) for creating each 4 KB type sort work file table space. For example, if you specify a value of 16 for the 4K SORT PRIMARY SPACE field and a value of 4 for the 4K SORT WORK TS’S field, the PRIQTY for each 4 KB type table space is 4096 KB.

The secondary space quantity (SECQTY) value for each 4 KB type table space is always 0.

If you run the CLIST in MIGRATE mode, the value of the 4K SORT PRIMARY SPACE field is used only when additional 4 KB type sort work file table spaces are created. Db2 does not take into account the space that is already used by existing 4 KB type work files.
4K SORT WORK TS'S field
The value of the 4K SORT WORK TS'S field specifies the number of 4 KB sort
work file table spaces that are to be created.

Acceptable values: for installation: 1 - 499; for migration: 0 - 500
Default: for installation: 1 for migration: 0
Update: none
DSNZPxxx: none

The sum of the values of the 4K SORT WORK TS'S field, the 32K SORT WORK
TS'S field, the 4K TEMP WORK TS'S field, and the 32K TEMP WORK TS'S field
cannot exceed 500.

If you run the CLIST in MIGRATE mode, and you enter a non-zero value, the
value is used for creating new 4 KB type sort work file table spaces. Db2 does not
take into account the number of existing 4 KB type sort work file table spaces.

4K SORT WORK SEG SIZE field
The value of the 4K SORT WORK SEG SIZE field specifies the segment size that is
to be used for 4 KB sort work file table spaces. The value must be a multiple of 4
between 4 and 64, inclusive.

Acceptable values: 4, 8, 12, ..., 60, 64 (multiples of 4)
Default: 16
Update: none
DSNZPxxx: none

The value of the 4K SORT WORK SEG SIZE field does not apply to existing 4 KB
work file table spaces.

32K SORT PRIMARY SPACE field
The value of the 32K SORT PRIMARY SPACE field specifies the total amount of
space for all 32 KB sort work table spaces in the work file database. This value is
specified in MB but you can enter the value in GB by specifying a suffix of G, for
example 1024G.

Unlike temporary work table spaces, sort work table spaces are always segmented
table spaces and cannot increase in size.

Acceptable values: for installation: 1 - 32,702,464; for migration: 0 - 32,702,464
Default: for installation: 20; for migration: 0
Update: none
DSNZPxxx: none

The value of the 32K SORT PRIMARY SPACE field is divided by the value of the
32K SORT WORK TS'S field to determine the primary space quantity (PRIQTY) for
creating each 32 KB type sort work file table space. For example, if you specify a
value of 16 for the 32K SORT PRIMARY SPACE field and a value of 4 for the 32K
SORT WORK TS'S field, the PRIQTY for each 32 KB type table space is 4096 KB.

The secondary space quantity (SECQTY) value for each 32 KB type table space is
always 0.
If you run the CLIST in MIGRATE mode, the value of the 32K SORT PRIMARY SPACE field is used only when additional 32 KB type sort work file table spaces are created. Db2 does not take into account the space that is already used by existing 32 KB type work files.

**32K SORT WORK TS’S field**
The value of the 32K SORT WORK TS’S field specifies the number of 32 KB sort work file table spaces that are to be created.

Acceptable values: for installation: 1 - 499; for migration: 0 - 500
Default: for installation: 1 for migration: 0
Update: none
DSNZPxxx: none

The sum of the values of the 4K SORT WORK TS’S field, the 32K SORT WORK TS’S field, the 4K TEMP WORK TS’S field, and the 32K TEMP WORK TS’S field cannot exceed 500.

If you run the CLIST in MIGRATE mode, and you enter a non-zero value, the value is used for creating new 32 KB type sort work file table spaces. Db2 does not take into account the number of existing 32 KB type sort work file table spaces.

**32K SORT WORK SEG SIZE field**
The value of the 32K SORT WORK SEG SIZE field specifies the segment size that is to be used for 32 KB sort work file table spaces. The value must be a multiple of 4 between 4 and 64, inclusive.

Acceptable values: 4, 8, 12, ..., 60, 64 (multiples of 4)
Default: 16
Update: none
DSNZPxxx: none

The value of the 32K SORT WORK SEG SIZE field does not apply to existing 32 KB work file table spaces.

**4K TEMP PRIMARY SPACE field**
The value of the 4K TEMP PRIMARY SPACE field specifies the total amount of space for all 4 KB temporary work table spaces in the work file database. This value is specified in MB, but you can enter the value in GB by specifying a suffix of G, for example 1024G.

Unlike sort work table spaces, temporary work table spaces can increase in size beyond the primary allocation. In installation mode, temporary work table spaces, unlike sort work table spaces, are created as either partition-by-growth table spaces or segmented table spaces, depending on the amount of primary space per table space.

Acceptable values: for installation: 1 - 8,388,608,000; for migration: 0 - 32,702,464
Default: for installation: 20; for migration: 0
Update: none
DSNZPxxx: none

The value of the 4K TEMP PRIMARY SPACE field is divided by the value of the 4K TEMP WORK TS’S field to determine either of the following values:
The primary space quantity (PRIQTY)

The maximum number of partitions (MAXPARTITIONS) and the maximum size (DSSIZE) of each partition

The PRIQTY value is used for creating each 4 KB type temporary work table space as a segmented table space. The MAXPARTITIONS and DSSIZE values are used for creating each 4 KB type temporary work table space as a partition-by-growth table space.

When there is 1 GB or less per table space, segmented table spaces are created by using the PRIQTY parameter. Otherwise, partition-by-growth table spaces are created by using the DSSIZE and MAXPARTITIONS parameters. If necessary, the CLIST curtails the calculated space at 16,384 GB, which is the maximum for a 4 KB page size partition-by-growth table space.

For example:

- If you specify a value of 16 for the 4K TEMP PRIMARY SPACE field and a value of 4 for the 4K TEMP WORK TS’S field, the PRIQTY for each 4 KB type table space is 4096 KB.
- If you specify a value of 16G for the 4K TEMP PRIMARY SPACE field and a value of 4 for the 4K TEMP WORK TS’S field, the MAXPARTITIONS for each 4 KB type table space is 1 with a DSSIZE of 4.

If you run the CLIST in MIGRATE mode, the value of the 4K TEMP PRIMARY SPACE field is used only when additional 4 KB type work file table spaces are created. Db2 does not take into account the space that is already used by existing 4 KB type work files.

4K TEMP WORK TS’S field

The value of the 4K TEMP WORK TS’S field specifies the number of 4 KB temporary work file table spaces that are to be created.

Acceptable values: 1 - 499; for migration: 0 - 500
Default: 1; for migration: 0
Update: none
DSNZPxxx: none

The sum of the values of the 4K SORT WORK TS’S field, the 32K SORT WORK TS’S field, the 4K TEMP WORK TS’S field, and the 32K TEMP WORK TS’S field cannot exceed 500.

If you run the CLIST in MIGRATE mode, and you enter a non-zero value, the value is used when new 4 KB type temporary work file table spaces are created. Db2 does not take into account the number of existing 4 KB type temporary work file spaces.

4K TEMP WORK SEG SIZE field

The value of the 4K TEMP WORK SEG SIZE field specifies the segment size that is to be used for 4 KB temporary work file table spaces. The value must be a multiple of 4 between 4 and 64, inclusive.

Acceptable values: 4, 8, 12, ..., 60, 64 (multiples of 4)
Default: 16
Update: none
DSNZPxxx: none
32K TEMP PRIMARY SPACE field

The value of the 32K TEMP PRIMARY SPACE field specifies the total amount of space for all 32 KB temporary work table spaces in the work file database. This value is specified in MB, but you can enter the value in GB by specifying a suffix of G, for example 1024G.

Unlike sort work table spaces, temporary work table spaces can increase in size beyond the primary allocation. In installation mode, temporary work table spaces, unlike sort work table spaces, are created as either partition-by-growth or segmented table spaces, depending on the amount of primary space per table space.

Default: for installation: 20; for migration: 0
Update: none
DSNZPxxx: none

The value of the 32K TEMP PRIMARY SPACE field is divided by the value of the 32K TEMP WORK TS'S field to determine either of the following values:

- The primary space quantity (PRIQTY)
- The maximum number of partitions (MAXPARTITIONS) and the maximum size (DSSIZE) of each partition

The PRIQTY value is used for creating each 32 KB type temporary work table space as a segmented table space. The MAXPARTITIONS and DSSIZE values are used for creating each 32 KB type temporary work table space as a partition-by-growth table space.

When there is 1 GB or less per table space, segmented table spaces are created by using the PRIQTY parameter. Otherwise, partition-by-growth table spaces are created using the DSSIZE and MAXPARTITIONS parameters. If necessary, the CLIST curtails the calculated space at 131,072 GB, which is the maximum for a 32 KB page size partition-by-growth table space.

For table spaces that are larger than 16,384 GB, data sets must be associated with a Data Facility Storage Management Subsystem (DFSMS) data class that has been specified with extended format and extended addressability.

For example:

- If you specify a value of 16 for the 32K TEMP PRIMARY SPACE field and a value of 4 for the 32K TEMP WORK TS’S field, the PRIQTY value for each 32 KB type table space is 4096 KB.
- If you specify a value of 16G for the 32K TEMP PRIMARY SPACE field and a value of 4 for the 32K TEMP WORK TS’S field, the MAXPARTITIONS value for each 32 KB type table space is 1 and the DSSIZE is 4.

If you run the CLIST in MIGRATE mode, the value of the 32K TEMP PRIMARY SPACE field is used only when additional 32 KB type work file table spaces or partition-by-growth table spaces are created. Db2 does not take into account the space that is already used by existing 32 KB type work files or partition-by-growth table spaces. If necessary, the CLIST curtails the calculated space at 131,072 GB, which is the maximum for a 32 KB page size partition-by-growth table space.
**32K TEMP WORK TS’S field**

The value of the 32K TEMP WORK TS’S field specifies the number of 32 KB temporary work file table spaces that are to be created.

Acceptable values: for installation: 1 - 499; for migration: 0 - 500
Default: for installation: 1; for migration: 0
Update: none
DSNZPxxx: none

The sum of the values of the 4K SORT WORK TS’S field, the 32K SORT WORK TS’S field, the 4K TEMP WORK TS’S field, and the 32K TEMP WORK TS’S field cannot exceed 500.

If you run the CLIST in INSTALL mode, you must enter a value of at least 1.

If you run the CLIST in INSTALL mode, and you do not have an existing 32 KB work file table space, enter a non-zero value. If you enter a non-zero value, the value is used for creating new 32 KB type temporary work file table spaces. Db2 does not take into account the number of existing 32 KB type temporary work file table spaces.

**32K TEMP WORK SEG SIZE field**

The value of the 32K TEMP WORK SEG SIZE field specifies the segment size that is to be used for 32 KB temporary work file table spaces. The value must be a multiple of 4 between 4 and 64, inclusive.

Acceptable values: 4, 8, 12, ..., 60, 64 (multiples of 4)
Default: 16
Update: none
DSNZPxxx: none

**DSNTIP91: Work file database panel 2**

The fields on the DSNTIP91 panel configure options for managing the Db2 work file database and thresholds for reporting excess work file space usage.

*Updating the parameters:* You can use the UPDATE mode of the installation CLIST to update any value on this panel.
Related tasks:

- Creating additional work file table spaces to reduce contention (Db2 Performance)

**MAX TEMP STG/AGENT field (MAXTEMPS subsystem parameter)**

The MAXTEMPS subsystem parameter determines the maximum amount of temporary storage in the work file database that a single agent can use at any given time for any type of usage. The usage can be for sort work files, created global temporary tables, declared global temporary tables, scrollable cursors result tables, and so on.

- **Acceptable values:** 0 - 2147483647
- **Default:** 0
- **Update:** option 18 on panel DSNTIPB
- **DSNZPxxx:** DSN6SPRM MAXTEMPS

An entry of 0 means that no limit is enforced.

By default, the specified value for MAXTEMPS is in MB, and in job DSNTIJUZ, the parameter value must be specified in MB. However, you can enter a GB value in the MAX TEMP STG/AGENT field of the installation CLIST by specifying a suffix of G, for example 46.

**SEPARATE WORK FILES field (WFDBSEP subsystem parameter)**

The WFDBSEP subsystem parameter specifies whether Db2 should provide an unconditional separation of table spaces in the work file database based on the allocation attributes of the table spaces.

- **Acceptable values:** YES, NO
- **Default:** NO
- **Update:** option 18 on panel DSNTIPB
- **DSNZPxxx:** DSN6SPRM.WFDBSEP
YES

Db2 directs declared global temporary table work only to Db2-managed
(STOGROUP) work file table spaces that are defined as partition-by-growth
(regardless of the SECQTY setting) or segmented (non-universal) table spaces
with a non-zero SECQTY setting. Db2 directs work file work only to
Db2-managed table spaces that are defined as segmented (non-universal) with
a zero SECQTY value or user-managed table spaces. If no table space with the
preferred allocation type is available, Db2 issues an error message, a negative
SQLCODE, or both.

NO

Db2 attempts to direct declared global temporary table work to Db2-managed
(STOGROUP) work file table spaces that are defined as partition-by-growth
(regardless of the SECQTY value). It then attempts to direct the work to
segmented (non-universal) table spaces with a non-zero SECQTY value. Db2
attempts to direct work file work to Db2-managed table spaces that are defined
as segmented (non-universal) with a zero SECQTY value or user-managed
table spaces. If no table space with the preferred allocation type is available,
Db2 selects a table space with a non-preferred allocation type.

MAX TEMP RID field (MAXTEMPS_RID subsystem parameter)

The MAXTEMPS_RID subsystem parameter determines the maximum amount of
temporary storage in the work file database that a single RID list can use at a time.

The work file storage is used for the RID list when the RID pool storage cannot
be used to contain all the RIDs. In other words, when RID pool storage overflow
occurs for a RID list, the Db2 subsystem attempts to store the RID list in work file
storage instead of falling back to a relational scan. This subsystem parameter
specifies the maximum number of RIDs (measured in RID blocks) that the
subsystem is allowed to store in the work file. If the maximum number of RIDs is
exceeded, the subsystem falls back to a relational scan.

The MAXTEMPS_RID subsystem parameter does not affect RID list processing for
pair-wise join.

Acceptable values: NONE, NOLIMIT, or 1 to 329166
Default: NOLIMIT
Update: option 18 on panel DSNTIPB
DSNZPxxx: MAXTEMPS_RID

The MAXTEMPS_RID value is specified in number of RID blocks. An entry of
NONE means that the RID list processing will not use work file storage. An entry
of NOLIMIT means that the number of RID blocks that can be stored in a work file
is unrestricted. This is the default behavior.

Each RID block that is stored in the work file occupies 32 KB of work file storage
and contains 4077 RIDs. For example, a value of 10000 for MAXTEMPS_RID limits
the number of RIDs that are allowed to be stored in the work file to 40,770,000, or
approximately 312.5 MB.

Another subsystem parameter, MAXTEMPS, controls the maximum amount of
temporary storage in the work file database that a single agent can use at a time
for any type of usage. The scope of the parameter includes work file usage for
storing RIDs. Therefore, if MAXTEMPS is set to a non-zero value, it overrides the
setting of MAXTEMPS_RID in some cases.

For example, consider the following scenario:
MAXTEMPS is set to 2G, meaning that 2 GB of work file storage is allowed to be used by the agent. MAXTEMPS_RID is set to 20000, meaning that 20,000 RID blocks are allowed to be stored in a work file, which translates to 625 MB of work file storage. Suppose that a RID list processing encountered a RID pool overflow, and therefore, the Db2 subsystem is using work file storage to store the RID list. Suppose that 65,328,000 RIDs are fetched into the RID list. The list therefore, takes up 16,000 RID blocks, which is below the MAXTEMPS_RID value of 20000. If no other activity is using much work file storage in the agent, the RID list should be successfully stored in a work file, and RID processing could continue. However, if other activities in the agent are using more than 1.5 GB of work file storage, less than 500 MB of work file storage is available to the agent (MAXTEMPS - 1.5 GB). The 16,000 RID blocks cannot fit in the 500 MB of available space. RID processing will not have enough work file storage and will fall back to a relational scan although the MAXTEMPS_RID value is well above the amount that the RID list needs.

Related tasks:

- Managing RID pool size (Db2 Performance)

Related reference:

- RID POOL SIZE field (MAXRBLK subsystem parameter)
- MAX TEMP STG/AGENT field (MAXTEMPS subsystem parameter)

**AGENT LEVEL THRESHOLD field (WFSTGUSE_AGENT_THRESHOLD subsystem parameter)**

The WFSTGUSE_AGENT_THRESHOLD subsystem parameter determines the percentage of available space in the work file database on a Db2 subsystem or data sharing member that can be consumed by a single agent before a warning message is issued.

Space in the work file database can be configured for declared temporary table work and for sort work. Sort work includes work for global temporary tables and transition tables for triggers.

Acceptable values: 0 - 100
Default: 0
Update: option 18 on panel DSNTIPB
DSNZPxxx: WFSTGUSE_AGENT_THRESHOLD

When this value is 0, Db2 does not issue agent-level space usage warnings for the work file database.

When this value is greater than 0, and subsystem parameter WFDBSEP is set to YES, Db2 issues a warning message when the percentage of total temporary work file space or the percentage of total sort work file space that is consumed by an agent exceeds this value.

When this value is greater than 0, and subsystem parameter WFDBSEP is set to NO, Db2 issues a warning message when the percentage of all work file space that is consumed by an agent exceeds this value.
**SYSTEM LEVEL THRESHOLD field**  
(WFSTGUSE_SYSTEM_THRESHOLD subsystem parameter)

The WFSTGUSE_SYSTEM_THRESHOLD subsystem parameter determines the percentage of available space in the work file database on a Db2 subsystem or data sharing member that can be consumed by all agents before a warning message is issued.

Space in the work file database can be configured for declared temporary table work and for sort work. Sort work includes work for global temporary tables and transition tables for triggers.

Acceptable values: 0 - 100  
Default: 90  
Update: option 18 on panel DSNTIPB  
DSNZPxxx: WFSTGUSE_SYSTEM_THRESHOLD

When this value is 0, Db2 does not issue system-level space usage warnings for the work file database.

When this value is greater than 0, and subsystem parameter WFDBSEP is set to YES, Db2 issues a warning message when the percentage of total temporary work file space or the percentage of total sort work file space that is consumed by all agents exceeds this value.

When this value is greater than 0, and subsystem parameter WFDBSEP is set to NO, Db2 issues a warning message when the percentage of all work file space that is consumed by all agents exceeds this value.

**DSNTIPE: Thread management panel 1**

The entries on the DSNTIPE panel determine main storage sizes.

*Updating the parameters:* You can use the UPDATE mode of the CLIST to update any value on this panel.

---

<table>
<thead>
<tr>
<th>DSNTIPE</th>
<th>INSTALL DB2 - THREAD MANAGEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;&gt;&gt; _</td>
<td></td>
</tr>
<tr>
<td>Check numbers and reenter to change:</td>
<td></td>
</tr>
<tr>
<td>1 DATABASES</td>
<td>&gt;&gt;&gt; 200</td>
</tr>
<tr>
<td>2 MAX USERS</td>
<td>&gt;&gt;&gt; 200</td>
</tr>
<tr>
<td>3 MAX REMOTE ACTIVE</td>
<td>&gt;&gt;&gt; 200</td>
</tr>
<tr>
<td>4 MAX REMOTE CONNECTED</td>
<td>&gt;&gt;&gt; 10000</td>
</tr>
<tr>
<td>5 MAX TSO CONNECT</td>
<td>&gt;&gt;&gt; 50</td>
</tr>
<tr>
<td>6 MAX BATCH CONNECT</td>
<td>&gt;&gt;&gt; 50</td>
</tr>
<tr>
<td>7 MAX KEPT DYN STMTS</td>
<td>&gt;&gt;&gt; 5000</td>
</tr>
<tr>
<td>8 MAX OPEN FILE REFs</td>
<td>&gt;&gt;&gt; 100</td>
</tr>
</tbody>
</table>

PRESS: ENTER to continue  RETURN to exit  HELP for more information

---

*Figure 23. Thread management panel: DSNTIPE*
**DATABASES field for panel DSNTIPE**

The value of the DATABASES field of panel DSNTIPE indicates the maximum number of databases that are to be open at one time.

- **Acceptable values:** 1 - 800
- **Default:** 200
- **Update:** option 19 on panel DSNTIPB
- **DSNZPxxx:** none

The maximum number of concurrent databases is affected primarily by the DSMAK field on panel DSNTIPC, which specifies the maximum number of open data sets.

**Related tasks:**
- [Programming applications for performance (Db2 Performance)](http://www.example.com)

**Related reference:**
- [DSMAX field (DSMAX subsystem parameter)](http://www.example.com)

**MAX USERS field (CTHREAD subsystem parameter)**

The CTHREAD subsystem parameter controls the maximum number of allied threads that are to be allocated concurrently. *Allied threads* are threads that are started at the local subsystem.

- **Acceptable values:** 1 - 20000
- **Default:** 200
- **Update:** option 19 on panel DSNTIPB
- **DSNZPxxx:** DSN6SYS CTHREAD

Count each of the following items as separate users:

- Each TSO user (whether running a DSN command or a Db2 request from QMF)
- Each batch job (whether running a DSN command or a Db2 utility)
- Each IMS region that can access Db2
- Each active CICS transaction that can access Db2
- Each utility (each utility uses one thread, plus one thread for each subtask)
- Each connection from users of CAF and RRSAF
- Each WebSphere application server or other application server that uses JDBC or ODBC has multiple connections to Db2. The number of connections is determined by the application server.

The maximum number of threads accessing data that can be allocated concurrently is the sum of the value of the MAX USERS field and the MAX REMOTE ACTIVE field on panel DSNTIPE. The maximum allowable value for this sum is 20000. When the number of users who are attempting to access Db2 exceeds the number that you specify, excess plan allocation requests are queued. In most situations, the amount of real storage determines the maximum number of threads that Db2 can handle.

Due to parallelism, Db2 utilities each use a minimum of one thread, plus an additional thread for each subtask. Therefore, a single utility might use many threads. Specify a thread value accordingly to accommodate parallelism within utilities. Consider using a value that is higher than the default value or the value that you specified in a previous version of Db2.
MAX REMOTE ACTIVE field (MAXDBAT subsystem parameter)
The value of the MAXDBAT subsystem parameter specifies the maximum number of database access threads (DBATs) that are allowed to be concurrently active.

Acceptable values: 0 - 19999
Default: 200
Update: option 19 on panel DSNTIPB

The total number of threads accessing data concurrently is the sum of the value of the MAX USERS field and this field, MAX REMOTE ACTIVE. The maximum allowable value for this sum is 20000. If a request for a new connection to Db2 is received and MAX REMOTE ACTIVE has been reached, the resulting action depends on whether ACTIVE or INACTIVE is specified for the DDF THREADS field on panel DSNTIPR.

If DDF THREADS is ...
ACTIVE
The allocation request is allowed but any further processing for the connection is queued waiting for an active database access thread to terminate.

INACTIVE
The allocation request is allowed and is processed when Db2 can assign an unused database access thread slot to the connection.

Setting MAX REMOTE ACTIVE to 0
You can use a 0 in this field to restrict DDF server activity on a member of a data sharing group. A setting of 0 produces the following results:

• DDF does not register the member with WLM for member-specific Sysplex routing. This does not prevent the member from using WLM for enclave prioritization, but it prevents WLM from including this member in the Sysplex routing data that is sent to remote sites.

• In an SNA network, DDF does not register the member’s LU name with the VTAM generic LU name during DDF startup. This causes VTAM generic resource connections to be directed to Db2 members that specify a MAX REMOTE ACTIVE value of greater than 0.

• In a TCP/IP network, DDF does not listen on the DRDA SQL port. This means TCP/IP SQL connections can be accepted only by members that specify MAX REMOTE ACTIVE greater than 0.

• DDF rejects requests for the Sysplex Routing TPN with the following sense code: SNA TPN not available.

• MAX REMOTE CONNECTED (CONDBAT) will be lowered to zero by default if it is not set explicitly.

• Db2 will still accept inbound Automatic Resynchronization requests.

MAX REMOTE CONNECTED field (CONDBAT subsystem parameter)
The CONDBAT subsystem parameter sets the maximum number of concurrent inbound DDF connections that are to be allowed.

Acceptable values: 0 - 150000
Default: 10000
Update: option 19 on panel DSNTIPB
This value must be greater than or equal to the value of MAX REMOTE ACTIVE. If MAX REMOTE ACTIVE is set to zero, MAX REMOTE CONNECTED is also set to zero. If a request to allocate a new connection to Db2 is received, and MAX REMOTE CONNECTED has been reached or MAX REMOTE CONNECTED is zero, the connection request is rejected.

Related tasks:
- Setting thread limits for database access threads (Db2 Performance)

MAX TSO CONNECT field (IDFORE subsystem parameter)
The IDFORE subsystem parameter determines the maximum number of users that can be identified to Db2 from TSO foreground at the same time.

Acceptable values: 1 - 20000
Default: 50
Update: option 19 on panel DSNTIPB

Count each of the following items as a separate user:
- Each TSO foreground user that is executing a DSN command.
- Each TSO foreground user that is connected to Db2 through the CAF or RRSAF. This can include QMF users who are running in TSO foreground or user-written CAF or RRSAF applications running in TSO foreground.

When the number of TSO users who are attempting to access Db2 exceeds the number you specify, excess connection requests are rejected. No Db2 subsystem parameter controls the maximum concurrent connections for IMS and CICS. You can control those limits by using IMS and CICS facilities. For the CICS attachment, the maximum number of connections to Db2 by using the resource control table (RCT) TYPE=INIT THRDMAX value.

MAX BATCH CONNECT field (IDBACK subsystem parameter)
The IDBACK subsystem parameter determines the maximum number of concurrent connections that can be identified to Db2 from batch.

Acceptable values: 1 - 20000
Default: 50
Update: option 19 on panel DSNTIPB

Recommendation: Use a minimum setting of 30.

Count each of the following items as a separate connection:
- Each Db2 utility.
- Each batch job that uses QMF.
- Each batch job that uses the DSN command processor.
- Each task that is connected to Db2 through the call attachment facility, which runs in batch. Among others, this can include:
  - Batch jobs that use QMF.
  - TCP/IP FTP connections.
• Each RRSAF connection that runs in batch.
• Each WebSphere application server or other application server that uses JDBC or ODBC has multiple connections to Db2. The number of connections is determined by the application server.

Batch job requests to access Db2 that exceed this limit are rejected.

REBUILD INDEX processing uses Db2 connections and might cause message DSNU397I to be issued. If you receive message DSNU397I indicating the REBUILD INDEX utility is constrained, increase the number of concurrent connections.

**MAX KEPT DYN STMTS field (MAXKEEPD subsystem parameter)**

The MAXKEEPD subsystem parameter specifies the maximum number of prepared, dynamic SQL statements that are to be saved past the point of commit or rollback. This parameter applies to applications that run with the KEEPDYNAMIC(YES) bind option.

Acceptable values: 0 - 204800
Default: 5000
Update: option 19 on panel DSNTIPB
DSNZPxxx: DSN6SPRM MAXKEEPD

This is a system-wide limit. This parameter does not limit the size of the dynamic cache itself.

When many applications that are bound with KEEPDYNAMIC(YES) run in a system that has the dynamic statement cache active, they can use a considerable amount of storage in the DBM1 address space. This parameter helps limit the amount of storage that these applications use by limiting the total number of prepared statements held by these applications past the point of commit or rollback. If this limit is exceeded, Db2 honors the KEEPDYNAMIC(YES) behavior, but "implicit" prepares might be necessary to rebuild the executable version of some SQL statements when they are executed after the point of commit or rollback.

If you enter 0, Db2 cannot keep the executable version of dynamic SQL statements past the point of commit or rollback. To retain the KEEPDYNAMIC(YES) behavior after the point of commit or rollback, Db2 implicitly prepares dynamic SQL statements to rebuild the executable version.

Related concepts:
- Methods for keeping prepared statements after the point of commit or rollback (Db2 Performance)

Related tasks:
- Minimizing the use of real and virtual storage (Db2 Performance)

Related reference:
- KEEPDYNAMIC bind option (Db2 Commands)

**MAX OPEN FILE REFS field (MAXOFILR subsystem parameter)**

The MAXOFILR subsystem parameter specifies the maximum number of data sets that can be open concurrently for the processing of LOB file references.

Acceptable values: 0 to value specified in MAX USERS field
Default: 100
Update: option 19 on panel DSNTIPB
DSNTIPE1: Thread management panel

The entries on the DSNTIPE1 panel determine main storage sizes.

*Updating the parameters:* You can use the UPDATE mode of the CLIST to update any value on this panel.

**Figure 24.** Thread management panel 2: DSNTIPE1

<table>
<thead>
<tr>
<th>DSNTIPE1 INSTALL DB2 - THREAD MANAGEMENT PANEL 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check numbers and reenter to change:</td>
</tr>
<tr>
<td>1 MANAGE REAL STORAGE ==&gt; AUTO       Manage real storage usage (AUTO,OFF,ON)</td>
</tr>
<tr>
<td>2 LONG-RUNNING READER       ==&gt; 10 Minutes before read claim warning</td>
</tr>
<tr>
<td>3 DATA DEF TIMEOUT       ==&gt; 1 For unavailable resources (-254-254)</td>
</tr>
<tr>
<td>4 INDEX CLEANUP THREADS ==&gt; 10 To remove pseudo empty index pages &amp; pseudo deleted index entries (0-128)</td>
</tr>
</tbody>
</table>

PRESS: ENTER to continue  RETURN to exit  HELP for more information

**MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter):**

The REALSTORAGE_MANAGEMENT subsystem parameter specifies whether Db2 should manage real storage consumption.

Acceptable values: ON, OFF, AUTO
Default: AUTO
Update: option 20 on panel DSNTIPB
DSNZPxxx: DSN6SPRM REALSTORAGE_MANAGEMENT

**ON** Db2 always discards unused real storage frames. Discarding the frames results in some CPU overhead, and this option is intended for systems in which the availability of real storage is limited. This value would most likely be appropriate for LPARs that have many Db2 subsystems, such as a development LPAR.

**OFF** Db2 does not discard unused real storage frames until one of the following conditions is met:
- The LPAR had reached an auxiliary critical state.
- The total real and auxiliary storage has reached 80% of the value of the REALSTORAGE_MAX subsystem parameter.
AUTO

Db2 discards unused real storage frames when a significant amount of paging activity is detected. By discarding frames, Db2 tries to bring the system to a point where paging is limited or nonexistent. However, it might not be possible to bring the system to that point if other applications on the same LPAR cause the shortage of real storage frames.

LONG-RUNNING READER field (LRDRTTHLD subsystem parameter)
The LRDRTTHLD subsystem parameter specifies the number of minutes that an agent is to be allowed to hold a read claim. If a read claim is held longer than allowed, Db2 issues a warning message, DSNB260I, and writes a trace record to report it as a long-running reader.

Acceptable values: 0 - 1439 minutes
Default: 10
Update: option 20 on panel DSNTIPB
DSNZPxxx: DSN6SPRM LRDRTTHLD

If you specify a value of 0, Db2 will not report long-running readers.

Related information:

DSNB260I (Db2 Messages)

DATA DEF TIMEOUT field (DDLTOX subsystem parameter)
The DDLTOX subsystem parameter specifies how long a request for SQL data definition is to wait for a resource before timing out. This parameter is used for the processing of DDL statements and GRANT, REVOKE, and LOCK statements.

Acceptable values: 0 to 254
Default: 1
Update: option 20 on panel DSNTIPB
DSNZPxxx: DSN6SPRM DDLTOX

0 Db2 resolves the timeout or deadlock situation by returning an error for data definition, GRANT, REVOKE, and LOCK statements.

1 - 254
The actual timeout value is the product of the value of this subsystem parameter and the value of the IRLMRWT subsystem parameter. The default value of 1 means that the value of the IRLMRWT subsystem parameter is used as the timeout value.

Related reference:

RESOURCE TIMEOUT field (IRLMRWT subsystem parameter)

INDEX CLEANUP THREADS field (INDEX_CLEANUP_THREADS subsystem parameter)
Specifies the maximum number of threads that can be created to process the cleanup of pseudo-deleted index entries on this subsystem or data sharing member. Pseudo-deleted entries in an index are entries that are logically deleted but still physically present in the index.

Acceptable values: 0 - 128
Default: 10
Data sharing scope: Members are not required to use the same setting
Online changeable: Yes
Update: option 20 on panel DSNTIPB

DSNZPxxx: DSN6SPRM INDEX_CLEANUP_THREADS

This parameter works in conjunction with the SYSIBM.SYSINDEXCLEANUP catalog table, which controls cleanup processing of pseudo-deleted index entries.

The default setting is appropriate for most situations. However, a larger setting might be appropriate in cases where large indexed tables or large numbers of indexed tables are constantly modified. A smaller value might be appropriate in cases where thread count is severely constrained.

When INDEX_CLEANUP_THREADS is set to 0, no cleanup is performed by the subsystem or data sharing member regardless of the entries in the SYSIBM.SYSINDEXCLEANUP catalog table.

Related reference:

- SYSINDEXCLEANUP catalog table (Db2 SQL)

**DSNTIP1: Buffer pool sizes panel 1**

The DSNTIP1 panel is the first of two panels on which you can specify the size of your virtual buffer pools.

**Updating the default buffer pool assignments:** You can also change the default buffer pool assignments for user data on this panel. Objects in the catalog and directory always use the following buffer pools, and the assignments cannot be changed: BP0, BP8K0, BP16K0, and BP32K.

It is best to separate the buffer pools for Db2 catalog and directory from the buffer pools for user data to isolate catalog and directory activities. In most cases, choose buffer pools other than the default buffer pools for user data, user indexes, and work files.

For more information about default buffer pool assignments, see [Deciding how many buffer pools to use (Db2 Performance)](#).

**Updating the buffer pool sizes:** You can change your buffer pool sizes online with the ALTER BUFFERPOOL command, but you cannot change these sizes by running the DSNTINST CLIST in update mode.

**Restriction:** The total amount of storage that is available in all buffer pools cannot exceed 16 TB. The amount of storage that is configured on the subsystem can further limit the buffer pool sizes. If you specify more storage than the total real storage that is available, performance can be impacted.
Related tasks:

- Assigning database objects to buffer pools (Db2 Performance)
- Calculating buffer pool size
- Monitoring and tuning buffer pools by using online commands (Db2 Performance)

Related reference:

- DSNTIP2: Buffer pool sizes panel 2
- ALTER BUFFERPOOL (Db2) (Db2 Commands)

**DEFAULT 4-KB BUFFER POOL FOR USER DATA field**

(TBSBPOOL subsystem parameter)

The TBSBPOOL subsystem parameter specifies the default buffer pool to use for 4 KB page size table spaces that are created implicitly.

Acceptable values: Any 4 KB buffer pool names
Default: BP0
Update: option 21 on panel DSNTIPB

If you do not specify a buffer pool name, Db2 uses the TBSBPOOL value as the default buffer pool for 4 KB page size table spaces. Db2 uses the value differently for the following statements:

**CREATE DATABASE**

Db2 uses the TBSBPOOL value.

**CREATE TABLE**

If you do not specify a table space name, Db2 implicitly creates a table space and selects the buffer pool as described in [Implicitly defined table spaces (Db2 Administration Guide)](http://example.com).

Related concepts:

- How SQL statements are invoked (Db2 SQL)
DEFAULT 8-KB BUFFER POOL FOR USER DATA field (TBSBP8K subsystem parameter)
The TBSBP8K subsystem parameter specifies the default buffer pool to use for 8 KB page size table spaces that are created implicitly.

Acceptable values: Any 8 KB buffer pool names
Default: BP8K0
Update: option 21 on panel DSNTIPB
DSNZPxxx: DSN6SYSP TBSBP8K

If you do not specify a buffer pool name, Db2 uses the TBSBP8K value as the default buffer pool for 8 KB page size table spaces. If you do not specify a table space name for a CREATE TABLE statement, Db2 implicitly creates a table space and selects the buffer pool as described in Implicitly defined table spaces (Db2 Administration Guide).

DEFAULT 16-KB BUFFER POOL FOR USER DATA field (TBSBP16K subsystem parameter)
The TBSBP16K subsystem parameter specifies the default buffer pool to use for 16 KB page size table spaces that are created implicitly.

Acceptable values: Any 16 KB buffer pool names
Default: BP16K0
Update: option 21 on panel DSNTIPB
DSNZPxxx: DSN6SYSP TBSBP16K

If you do not specify a buffer pool name, Db2 uses the TBSBP16K value as the default buffer pool for 16 KB page size table spaces. If you do not specify a table space name for a CREATE TABLE statement, Db2 implicitly creates a table space and selects the buffer pool as described in Implicitly defined table spaces (Db2 Administration Guide).

DEFAULT 32-KB BUFFER POOL FOR USER DATA field (TBSBP32K subsystem parameter)
The TBSBP32K subsystem parameter specifies the default buffer pool to use for 32 KB page size table spaces that are created implicitly.

Acceptable values: Any 32 KB buffer pool names
Default: BP32K
Update: option 21 on panel DSNTIPB
DSNZPxxx: DSN6SYSP TBSBP32K

If you do not specify a buffer pool name, Db2 uses the TBSBP32K value as the default buffer pool for 32 KB page size table spaces. If you do not specify a table space name for a CREATE TABLE statement, Db2 implicitly creates a table space and selects the buffer pool as described in Implicitly defined table spaces (Db2 Administration Guide).

DEFAULT BUFFER POOL FOR USER LOB DATA field (TBSBPLOB subsystem parameter)
The TBSBPLOB subsystem parameter specifies the default buffer pool that is to be used for LOB table spaces that are created implicitly. This parameter also applies to LOB tables spaces that are created explicitly without the BUFFERPOOL clause.
Acceptable values: Any 4 KB, 8 KB, 16 KB, or 32 KB buffer pool names
Default: BP0
Update: option 21 on panel DSNTIPB
DSNZPxxx: DSN6SYSP TBSBFLOB

Use privilege is required on the specified buffer pool to use it by default.

Related concepts:

How SQL statements are invoked (Db2 SQL)

**DEFAULT BUFFER POOL FOR USER XML DATA field (TBSBPXML subsystem parameter)**
The TBSBPXML subsystem parameter specifies the default buffer pool that is to be used for XML table spaces that are created implicitly.

Acceptable values: Any 16 KB buffer pool name
Default: BP16K0
Update: option 21 on panel DSNTIPB
DSNZPxxx: DSN6SYSP TBSBPXML

Use privilege is required on the specified buffer pool to use it by default.

Related concepts:

How SQL statements are invoked (Db2 SQL)

**DEFAULT BUFFER POOL FOR USER INDEXES field (IDXBPOOL subsystem parameter)**
The IDXBPOOL subsystem parameter identifies the default buffer pool that is to be used for indexes on user data.

Db2 uses the IDXBPOOL value if a value for INDEXBP is not specified on a CREATE DATABASE statement. Db2 does not use this value for a CREATE INDEX statement without the BUFFERPOOL option. In that case, Db2 uses the default index buffer pool for the database.

Acceptable values: Any 4-KB, 8-KB, 16-KB, or 32-KB buffer pool names
Default: BP0
Update: option 21 on panel DSNTIPB
DSNZPxxx: DSN6SYSP IDXBPOOL

**BUFFERPOOL fields BP0 - BP29**
The values of fields BP0 - BP29 on panel DSNTIP1 specify the number of 4 KB buffers in each of those virtual buffer pools (BP0 - BP29).

Acceptable values: BP0: 2000 - 250000000
BP1 - BP29: 0 - 250000000
Default: BP0: 20000
BP1 - BP29: 0
Update: ALTER BUFFERPOOL command
DSNZPxxx: none
Restriction: The sum of the storage that is available in all buffer pools cannot exceed 16 TB. The amount of storage configured on the subsystem can further limit the buffer pool sizes. If you specify more storage than the total real storage that is available, performance can suffer.

**DSNTIP2: Buffer pool sizes panel 2**

The DSNTIP2 panel is the second of the two panels that enables you to specify the size of your virtual buffer pools.

The total size of all buffer pools cannot exceed 16TB.

*Updating the buffer pool sizes:* You can change your buffer pool sizes online with the ALTER BUFFERPOOL command. You cannot change these sizes by running the DSNTINST CLIST in update mode.

![Figure 26. Buffer pool sizes panel 2: DSNTIP2](image)

**Related tasks:**
- Deciding how many buffer pools to use (Db2 Performance)
- Calculating buffer pool size
- Monitoring and tuning buffer pools by using online commands (Db2 Performance)

**Related reference:**
- DSNTIP1: Buffer pool sizes panel 1
- ALTER BUFFERPOOL (Db2) (Db2 Commands)

**BUFFERPOOL fields BP30 - BP32K9**

The values of fields BP30 - BP32K9 on panel DSNTIP2 specify the number of buffers in each of those virtual buffer pools (BP30 - BP32K9).
Acceptable values:

- BP30 - BP49: 0 - 250000000
- BP8K0 - BP8K9: 0 - 125000000
- BP16K0 - BP16K9: 0 - 62500000
- BP32K: 250 - 31250000
- BP32K1 - BP32K9: 0 - 31250000

Default:

- BP8K0: 2000
- BP16K0: 500
- BP32K: 250
- All others: 0

Update: ALTER BUFFERPOOL command

DSNZPxxx: none

Restriction: The sum of the storage that is available in all buffer pools cannot exceed 16 TB. The amount of storage that is configured on the subsystem can further limit the buffer pool sizes.

**DSNTIPN: Tracing parameters panel**

The entries on the DSNTIPN panel affect the audit, global, accounting, and monitor traces and the checkpoint frequency.

Because space on this panel is limited, some input fields are not wide enough to display the entire contents of the field at one time. These fields are defined as ISPF scrollable fields, which let you see the field contents. A less-than symbol (⟨) or a greater-than (⟩) symbol denotes a scrollable field. A ⟨ symbol indicates that the field can be scrolled to the right, and a ⟨ symbol indicates that the field can be scrolled to the left. You can position your cursor in the field and use ISPF scroll commands LEFT and RIGHT to scroll through the field. To see the entire contents of the field at one time, type EXPAND in the command field, position your cursor in the scrollable field, and press Enter. (Alternatively, you can assign your PF keys to be the LEFT, RIGHT, and EXPAND commands.)

---

**Figure 27. Tracing panel: DSNTIPN**

- **1 AUDIT TRACE**: > Audit classes to start. NO,YES,list
- **2 TRACE AUTO START**: > Global classes to start. YES,NO,list
- **3 TRACE SIZE**: 64K + Trace table size in bytes. 4K-396K
- **4 SMF ACCOUNTING**: 1 + > Accounting classes to start. NO,YES,list
- **5 SMF STATISTICS**: YES + Statistics classes to start. NO,YES,list
- **6 STATISTICS TIME**: 1 + Time interval in minutes. 1-60
- **7 STATISTICS SYNC**: NO + Synchronization within the hour. NO,0-59
- **8 DATASET STATS TIME**: 5 + Time interval in minutes. 1-60
- **9 MONITOR TRACE**: NO + > Monitor classes to start. NO,YES,list
- **10 MONITOR SIZE**: 1M + Default monitor buffer size. 1M-64M
- **11 UNICODE IFCIDS**: NO + Include UNICODE data when writing IFCIDS
- **12 DDF/RRSAF ACCUM**: 10 + Rollup acting for DDF/RRSAF. NO, 2-64K
- **13 AGGREGATION FIELDS**: 0 + Rollup acting for aggregation fields
- **14 COMPRESSION SMF REC**: OFF + Compress trace records destined for SMF

---

**Related concepts:**

- DSNTIPN
- INSTALL DB2 - TRACING PARAMETERS
- DB2 TRACING PARAMETERS
- INSTALL DB2 - INSTALL"
AUDIT TRACE field (AUDITST subsystem parameter)

The AUDITST parameter controls whether the audit trace is to start automatically when Db2 is started. You can specify the classes for which the audit trace is to start.

Acceptable values: NO, YES, list of classes, * (asterisk)
Default: NO
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYP AUDITST
Security parameter: Yes

This is a scrollable field.

NO Audit trace is not started automatically. If the audit trace is to be used, it must be started with the START TRACE command.

YES The audit trace for the default class (class 1) starts whenever Db2 is started.

list of classes
The audit trace for the specified classes starts whenever Db2 is started. To specify classes, list the numbers (any integer from 1 to 32) separated by commas.

* (asterisk)
The audit trace starts for all classes whenever Db2 is started.

Note: This is a security-related parameter. Audit traces should be enabled to collect information about Db2 security controls to ensure that data access is allowed only for authorized users.

Recommendation: Either AUDITST or audit policies should be enabled at Db2 startup.

Related concepts:

TRACE AUTO START field (TRACSTR subsystem parameter)

The TRACSTR subsystem parameter specifies whether the global trace is to start automatically when Db2 is started. This parameter also specifies the classes for which the global trace is to automatically start.

Acceptable values: YES, NO, list of classes, an asterisk (*)
Default: NO
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYP TRACSTR

This is a scrollable field.

YES
Starts the global trace for the default classes (classes 1, 2, and 3) whenever Db2 is started, and it performs additional data consistency checks whenever a data or index is modified.

NO Specifies no automatic start of the global trace. If the global trace is to be used, it must be started with a special START TRACE command.
**list of classes**
Starts the global trace for the specified classes. Enter a list of class numbers (any integer from 1 - 32) separated by commas. Only classes 1 - 9 are defined by Db2.

**asterisk (*)**
Starts global trace for all classes.

The global trace is used to diagnose problems in Db2. Users with production systems that require high performance might consider turning off global trace. However, be aware that turning off global trace presents a serviceability exposure. In the event of a system failure, IBM Support might request that you turn on global trace and attempt to re-create the problem.

**Related concepts:**
- Db2 audit trace (Managing Security)

**Related tasks:**
- Minimizing the processing cost of Db2 traces (Db2 Performance)

**TRACE SIZE field (TRACTBL subsystem parameter)**
The TRACTBL subsystem parameter specifies the size, in bytes, of the RES trace table. This table is the default destination for the global trace records in Db2. Most trace records require 32 byte entries. Events with more than three data items require 64 byte entries.

Acceptable values: 4K to 396K
Default: 64K
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP TRACTBL

You can use the abbreviation K for multiples of 1024 bytes. The actual value is rounded up to a multiple of 4. If you use 50K, for example, the actual table size is 52 KB.

In the subsystem parameter, use a multiple of 4. For example, to get a 64 KB table, code TRACTBL=16.

**Related concepts:**
- Db2 audit trace (Managing Security)

**SMF ACCOUNTING field (SMFACCT subsystem parameter)**
The SMFACCT subsystem parameter specifies whether Db2 is to send accounting data to SMF automatically when Db2 is started. This parameter also specifies which classes are to be sent.

Acceptable values: NO, YES, list of classes, * (asterisk)
Default: 1,2,3,7,8
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP SMFACCT

This is a scrollable field.

**NO** Specifies no automatic start of classes.
YES
Starts the trace for the default class (class 1). You might also need to update
the SMFPRMxx member of SYS1.PARMLIB to permit SMF to write the records.

list of classes
Starts the specified classes. Enter a list of class numbers (any integer from 1 -
32), separated by commas. Only classes 1 - 5, 7, 8, and 10 are defined by Db2.

* (asterisk)
Starts all classes.

Related concepts:

Accounting trace (Db2 Performance)

Related tasks:

Installation step 9: Record Db2 data to SMF (optional)

SMF STATISTICS field (SMFSTAT subsystem parameter)
The SMFSTAT subsystem parameter specifies whether Db2 is to send statistical
data to SMF automatically when Db2 is started. This parameter also specifies
which classes are sent.

Acceptable values: NO, YES, list of classes, an asterisk (*)
Default: YES
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP SMFSTAT

This is a scrollable field.

NO Specifies no automatic start.

YES
Starts the trace for the default classes (classes 1, 3, 4, 5, and 6). You might also
need to update the SMFPRMxx member of SYS1.PARMLIB to permit SMF to
write the records.

list of classes
Starts the specified classes. Enter a list of class numbers (any integer from 1 -
32), separated by commas. Only classes 1 - 5 are defined by Db2.

asterisk (*)
Starts all classes.

Related tasks:

Installation step 9: Record Db2 data to SMF (optional)

Monitoring concurrency and locks (Db2 Performance)

STATISTICS TIME field (STATIME subsystem parameter)
The STATIME subsystem parameter specifies the time interval, in minutes, between
statistics collections. Statistics records are to be written at approximately the end of
this interval.

The STATIME subsystem parameter applies only to IFCIDs 0105, 0106, 0199, and
0365.

Acceptable values: 1 - 60
Default: 1
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP STATIME
STATISTICS SYNC field (SYNCVAL subsystem parameter)
The SYNCVAL subsystem parameter specifies whether Db2 statistics recording is to be synchronized with some part of the hour, such as 15 minutes past the hour.

In Db2 12, the SYNCVAL subsystem parameter applies only to IFCIDs 0105, 0106, and 0199.

Acceptable values: NO, 0 - 59
Default: NO
Update: option 23 on panel DSNTIPB
DSNZP.xxx: DSN6SYSP SYNCVAL

NO Specifies that statistics recording is not synchronized. NO is the default.

0 - 59
Db2 statistics recording interval is synchronized with the beginning of the hour (0 minutes past the hour) or with any number of minutes past the hour up to 59. This parameter has no effect if STATIME is greater than 60.

Example

You want the Db2 statistics recording interval to have a length of 15 minutes and to be synchronized with 15 minutes past the hour. Thus, Db2 statistics are recorded at 15, 30, 45, and 60 minutes past the hour. To establish this interval, you would specify the following:

STATIME=15
SYNCVAL=15

DATASET STATS TIME field (DSSTIME subsystem parameter)
The DSSTIME subsystem parameter specifies the time interval, in minutes, that is to occur between the resetting of data set statistics for online performance monitors.

Acceptable values: 1 - 60
Default: 5
Update: option 23 on panel DSNTIPB
DSNZP.xxx: DSN6SYSP DSSTIME

Online performance monitors can request Db2 data set statistics for the current interval with an IFI READS request for IFCID 0199.

MONITOR TRACE field (MON subsystem parameter)
The MON subsystem parameter specifies whether the monitor trace is to start automatically when Db2 is started. The trace can be started for all classes, a default class, or a specified list of classes.

Acceptable values: NO, YES, list of classes, an asterisk (*)
Default: NO
Update: option 23 on panel DSNTIPB
DSNZP.xxx: DSN6SYSP MON

This is a scrollable field.

NO Specifies no automatic start.
YES
Starts the trace for the default classes (class 1) whenever Db2 is started.

list of classes
Start the trace automatically for the specified classes. Enter a list of class numbers (any integer from 1 - 32), separated by commas. Only classes 1 - 8 are defined by Db2.

asterisk (*)
Starts the trace automatically for all classes.

Related concepts:
Monitor trace (Db2 Performance)

MONITOR SIZE field (MONSIZE subsystem parameter)
The MONSIZE subsystem parameter determines the default buffer size that is to be used for monitor traces when data is sent to monitor destinations.

Acceptable values: 1048576 - 67108864
Default: 1048576
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP MONSIZE

You can enter the value in bytes (for example, 4194304) or use the abbreviation K for kilobytes (for example, 4096K) or M for megabytes (for example, 4M).

UNICODE IFCIDS field (UIFCIDS subsystem parameter)
The UIFCIDS subsystem parameter specifies whether output from IFC records is to include Unicode information.

Acceptable values: NO, YES
Default: NO
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP UIFCIDS

Only a subset of the character fields are encoded in Unicode. These fields are identified in the IFCID record definition by a %U in the comment area to the right of the field declaration in the DSNDQWxx copy files. The remaining fields maintain the same encoding of previous releases.

DDF/RRSAF ACCUM field (ACCUMACC subsystem parameter)
The ACCUMACC parameter determines whether Db2 accounting data is to be accumulated by the user for DDF and RRSAF threads.

Acceptable values: NO, 2 - 65535
Default: 10
Update: option 23 on panel DSNTIPB
DSNZPxxx: DSN6SYSP ACCUMACC

NO Db2 writes an accounting record when a DDF thread is made inactive or when sign-on processing occurs for an RRSAF thread.

integer
Db2 writes an accounting record every integer accounting intervals for a given user identifier. integer is between 2 and 65535.
A user identifier is a concatenation of one or more values, which can include the user ID, application name or transaction name, and workstation name. Those values are set by DDF threads, or by RRSAF threads through the RRSAF SIGNON, AUTH SIGNON, and CONTEXT SIGNON functions. The possible concatenations are described in "AGGREGATION FIELDS field (ACCUMUID subsystem parameter)."

An accounting record might be written prior to the completion of integer accounting intervals for a given user in the following cases:

- An internal storage threshold is reached for the accounting rollup blocks.
- Db2 has not rolled data into an internal block in approximately 10 minutes.

### Related concepts:
- Accounting trace (Db2 Performance)

### Related tasks:
- Minimizing the volume of Db2 trace data (Db2 Performance)
- Minimizing the processing cost of Db2 traces (Db2 Performance)
- Monitoring for performance exceptions (Db2 Performance)

### Related reference:
- AGGREGATION FIELDS field (ACCUMUID subsystem parameter)

**AGGREGATION FIELDS field (ACCUMUID subsystem parameter)**

The ACCUMUID subsystem parameter determines the aggregation fields that are to be used for DDF and RRSAF accounting rollup.

- Acceptable values: 0 - 17
- Default: 0
- Update: option 23 on panel DSNTIPB
- DSNZPxxx: DSN6SYS ACCUMUID

The values are defined in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Rollup criteria</th>
<th>String of X'00' or string of X'40' considered for rollup?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>User ID, application name or transaction name, and workstation name</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>User ID</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>User application name or transaction name</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>User workstation name</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>User ID and application name or transaction name</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>User ID and workstation name</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>User application name or transaction name and workstation name</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>User ID, application name or transaction name, and workstation name</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>User ID and application name or transaction name</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>User ID and workstation name</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>User application name or transaction name and workstation name</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 23. Values for the aggregation fields that are used for DDF and RRSAF accounting rollup (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Rollup criteria</th>
<th>String of X'00' or string of X'40' considered for rollup?</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>User ID, application name or transaction name, and workstation name</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>User ID</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>User application name or transaction name</td>
<td>Yes</td>
</tr>
<tr>
<td>14</td>
<td>User workstation name</td>
<td>Yes</td>
</tr>
<tr>
<td>15</td>
<td>User ID and application name or transaction name</td>
<td>Yes</td>
</tr>
<tr>
<td>16</td>
<td>User ID and workstation name</td>
<td>Yes</td>
</tr>
<tr>
<td>17</td>
<td>User application name or transaction name and workstation name</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note:
1. At least one value in the set of criteria must have a value other than a string of X'00's or a string of X'40's.

For example, if a thread has the user ID set to a string of X'00' and the workstation name set to myws, the thread qualifies for rollup if ACCUMUID is set to 5, but not if ACCUMUID is set to 9.

Assume that ACCUMUID is set to 5. The threads in the following table all qualify for rollup.

Table 24. Values for user ID and workstation name

<table>
<thead>
<tr>
<th>User ID</th>
<th>Workstation name</th>
</tr>
</thead>
<tbody>
<tr>
<td>myuser</td>
<td>myws</td>
</tr>
<tr>
<td>myuser</td>
<td>A string of X'40'</td>
</tr>
<tr>
<td>A string of X'40'</td>
<td>myws</td>
</tr>
<tr>
<td>myuser</td>
<td>A string of X'00'</td>
</tr>
<tr>
<td>A string of X'00'</td>
<td>myws</td>
</tr>
</tbody>
</table>

The thread with user ID set to myuser and workstation name set to a string of X'40' and the thread with user ID set to myuser and workstation name set to a string or X'00' are included in the same rollup record. The thread with user ID set to a string of X'40' and workstation name set to myws and the thread with user ID set to a string of X'00' and workstation name set to myws are included in the same rollup record.

Db2 writes individual accounting threads for threads that do not meet the criteria for rollup.

The value of ACCUMUID is ignored if the value of ACCUMACC is set to NO.

Related concepts:
- Accounting trace (Db2 Performance)

Related tasks:
- Minimizing the volume of Db2 trace data (Db2 Performance)
- Minimizing the processing cost of Db2 traces (Db2 Performance)
- Monitoring for performance exceptions (Db2 Performance)
**COMPRESS SMF RECS field (SMFCOMP subsystem parameter)**

The SMFCOMP subsystem parameter specifies whether Db2 is to compress trace records that are to be written to SMF.

Acceptable values: OFF, ON
Default: OFF
Update: option 23 on panel DSNTPB
DSNZPxxx: DSN6SYSYP SMFCOMP

**OFF**

SMF trace records are not compressed.

**ON**

Trace records that are to be written to SMF are compressed. The z/OS compression service CSRCESRV compresses everything after the SMF header (SM100END, SM101END, or SM102END). See SM100, SM101, and SM102 for compression flags and the uncompressed size of compressed data.

**DSNTIPO: Operator functions panel**

The entries on the DSNTIPO panel affect various operator functions. These functions include write-to-operator route codes, automatic recall, and the maximum amount of CPU time that is to be allocated for a dynamic SQL statement.

```
<table>
<thead>
<tr>
<th>DSNTIPO</th>
<th>INSTALL DB2 - OPERATOR FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter data below:</td>
</tr>
<tr>
<td>1 WTO ROUTE CODES  ==&gt; 1</td>
<td>Routing codes for WTORs</td>
</tr>
<tr>
<td>2 RECALL DATABASE ==&gt; YES</td>
<td>Use DFHSN automatic recall. YES or NO</td>
</tr>
<tr>
<td>3 RECALL DELAY  ==&gt; 120</td>
<td>Seconds to wait for automatic recall</td>
</tr>
<tr>
<td>4 AUTO BIND     ==&gt; YES</td>
<td>Use automatic bind. YES, NO, or COEXIST</td>
</tr>
<tr>
<td>5 EXPLAIN PROCESSING ==&gt; YES</td>
<td>Explain allowed on autobind? YES or NO</td>
</tr>
<tr>
<td>6 DPROP SUPPORT ==&gt; 1</td>
<td>1=NO 2=ONLY 3=ANY</td>
</tr>
<tr>
<td>7 SITE TYPE     ==&gt; LOCALSITE</td>
<td>LOCALSITE or RECOVERYSITE</td>
</tr>
<tr>
<td>8 TRACKER SITE  ==&gt; NO</td>
<td>Tracker DB2 system. NO or YES</td>
</tr>
<tr>
<td>9 READ COPY2 ARCHIVE ==&gt; NO</td>
<td>Read COPY2 archives first. NO or YES</td>
</tr>
<tr>
<td>10 PROFILE AUTOSTART ==&gt; NO</td>
<td>Auto start profile manager: NO or YES</td>
</tr>
</tbody>
</table>
```

Figure 28. Operator functions panel: DSNTIPO

**WTO ROUTE CODES field (ROUTCDE subsystem parameter)**

The ROUTCDE subsystem parameter specifies the z/OS console routing codes that are to be assigned to messages that are not solicited from a specific console.

Acceptable values: 1 - 16 route codes. Specify at least one code. Separate multiple codes by commas only, with no blanks; for example: 1,3,5,7,9,10,11.
Default: 1
Update: option 24 on panel DSNTPB
DSNZPxxx: DSN6SYSYP ROUTCDE

**Related reference:**
- Routing Codes (MVS System Messages)
- Routing the Message (z/OS MVS Assembler Services Guide)
RECALL DATABASE field (RECALL subsystem parameter)
The RECALL subsystem parameter controls whether DFSMShsm automatic recall is to be performed for Db2 databases.

Acceptable values: YES, NO  
Default: YES  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RECALL

YES  
Indicates that DFSMShsm is invoked to automatically recall migrated table spaces.

NO  
Indicates that a Db2 table space that has been migrated is considered to be an unavailable resource. It must be recalled explicitly before Db2 can use it.

RECALL DELAY field (RECALLD subsystem parameter)
The RECALLD subsystem parameter specifies the maximum length of time, in seconds, that a program can be delayed for a DFSMShsm recall.

Acceptable values: 0 - 32767  
Default: 120  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM RECALLD

If the recall is not completed within the specified number of seconds, the program receives an error message indicating that the set is unavailable, but that a recall was initiated. If you use 0 and the RECALL DATABASE field is YES, the recall is performed asynchronously. This field is ignored if the RECALL DATABASE field is NO.

The RECALL DELAY option is not used when running a Db2 utility against a Db2-migrated data set.

AUTO BIND field (ABIND subsystem parameter)
The ABIND subsystem parameter specifies whether Db2 automatically rebinds packages and plans in certain situations at run time.

Acceptable values: YES, NO, COEXIST  
Default: YES  
Update: option 24 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM ABIND

Automatic rebinds can improve the availability and administration of packages and plans, by reducing the need for explicit rebinds. However, automatic binds can also harm performance because they require access to the Db2 catalog and directory, resource contention might cause them to fail, and they might result in access path regressions.

Tip: For best results enable automatic binds by specifying COEXIST in data-sharing environments and YES in non-data-sharing environments, but also take action to avoid migration-related automatic binds. For details, see Avoid automatic binds at migration to Db2 12.
**Important:** Apply APAR PI87675 to remove the possibility of repeating automatic remigration binds.

**COEXIST or YES**

Db2 automatically rebinds packages or plans at run time in the following conditions:

- A package that is marked invalid runs (VALID='N' in the SYSIBM.SYSPACKAGE catalog table).
- A package that was last bound in a release earlier than DB2 10 runs in Db2 12.
- In release coexistence or fallback, a package or plan that was last bound on Db2 12 runs again on Db2 11. After the automatic bind, the package or plan continues to run without automatic binds, even if it runs again on Db2 12. Packages and plans continue to run on Db2 12 as bound on Db2 11. The plan or package does not use any new features that Db2 12 introduces.

**NO**

Specifies that no automatic binds occur. If any of the following conditions are true when a package attempts to run, Db2 issues SQLCODE -908 SQLSTATE 23510, unless you explicitly rebind the package:

- A package is marked invalid (VALID='N' in the SYSIBM.SYSPACKAGE catalog table).
- A package that was last bound (explicitly or automatically) in Db2 12 attempts to run on Db2 11.

**Related concepts:**

- Automatic binds in coexistence
- Changes that invalidate packages (Db2 Application programming and SQL)

**Related information:**

**EXPLAIN PROCESSING field (ABEXP subsystem parameter)**

The ABEXP subsystem parameter controls whether EXPLAIN processing is to occur during an automatic rebind.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>YES, NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>YES</td>
</tr>
<tr>
<td>Update:</td>
<td>option 24 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6SPRM ABEXP</td>
</tr>
</tbody>
</table>

**YES**

EXPLAIN processing occurs during the automatic rebind of a plan or package when the bind option EXPLAIN(YES) is specified. If the PLAN_TABLE does not exist, the automatic rebind continues, but generates no EXPLAIN output. EXPLAIN processing does not occur during the automatic rebind of plans or packages with the bind option EXPLAIN(NO).

**NO**

EXPLAIN processing does not occur during the automatic rebind of a plan or package.

**DPROP SUPPORT field (EDPROP and CHGDC subsystem parameters)**

The value of the DPROP SUPPORT field determines whether IMS DataPropagator is to be used to propagate SQL changes to tables defined with DATA CAPTURE CHANGES.

| Acceptable values: | 1, 2, 3 |
Default: 1
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6SPRM EDPROP, DSN6SPRM CHGDC

1 Specifies that you do not intend to propagate SQL changes.

2 Specifies that you intend to use IMS DataPropagator to propagate SQL changes, and that changes made to tables defined with DATA CAPTURE CHANGES are allowed only when all of the following conditions are met:
   - Monitor trace class 6 is active.
   - IMS DataPropagator is installed.
   - The Db2 application is running in an IMS environment.

If you specify 2 and monitor trace class 6 is not active, IMS DataPropagator is not installed, or the IMS DataPropagator application is not running in an IMS environment, then no changes to the Db2 table are permitted.

3 Specifies that data propagation occurs when all of the following conditions are met:
   - Monitor trace class 6 is active.
   - IMS DataPropagator is installed.
   - The Db2 application is running in an IMS environment.

The ANY option for IMS DataPropagator Support is intended for subsystems that need to propagate some data with IMS DataPropagator and need to propagate some data with a different program.

If you specify 3, an application that is not running in an IMS environment can update Db2 tables that are defined with DATA CAPTURE CHANGES. However, these changes are not propagated to IMS. You might have tables in your Db2-IMS environment that you want to be updated only by Db2 applications. You can protect these tables by using any of the following methods:
   - Use the ENABLE parameter on BIND to specify a specific attachment facility through which updates to data propagation tables can be made.
   - Define a validation procedure for data propagation tables to allow only certain plans to update those tables.
   - Use a group authorization ID to assign update authority for data propagation tables to a group of authorization IDs that can run only in the IMS environment.

The DPROP SUPPORT field values correspond with the CHGDC and EDPROP subsystem parameters as indicated in the following table.

<table>
<thead>
<tr>
<th>DPROP SUPPORT field value</th>
<th>CHGDC setting</th>
<th>EDPROP setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>2</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>3</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

Related concepts:

IMS DataPropagator
SITE TYPE field (SITETYP subsystem parameter)

The SITETYP subsystem parameter specifies whether the current system is at a local site or a recovery site.

Acceptable values: LOCALSITE, RECOVERYSITE
Default: LOCALSITE
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SITETYP

LOCALSITE
Specifies that the current system is at the site where the multiple image copies are made and are operational.

RECOVERYSITE
Specifies that the current system is at the site that is named as an alternative for recovery purposes.

The RECOVER utility uses this value to determine what site the current system is on. It then recovers everything from the copies of data that are registered at that site. The RECOVER and MERGECOPY utilities use this value to determine whether COPYDDN or RECOVERDDN is allowed with NEWCOPY NO.

TRACKER SITE field (TRKRSITE subsystem parameter)

The value of the TRKRSITE subsystem parameter indicates whether the subsystem that is being installed is to be used as a remote tracker site. A remote tracker site is used by another Db2 subsystem in case of a disaster.

Acceptable values: NO, YES
Default: NO
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6SPRM TRKRSITE

Related tasks:
- Recovering from disasters by using a tracker site (Db2 Administration Guide)

READ COPY2 ARCHIVE field (ARC2FRST subsystem parameter)

The ARC2FRST subsystem parameter determines whether COPY2 archives are to be read first when the Db2 subsystem is started.

Acceptable values: NO, YES
Default: NO
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6LOGP ARC2FRST

PROFILE AUTOSTART field (PROFILE_AUTOSTART subsystem parameter)

The PROFILE_AUTOSTART subsystem parameter specifies whether START PROFILE command processing is automatically initiated as part of Db2 start processing.

Acceptable values: NO, YES
Default: NO
Online changeable: no
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6SYSP PROFILE_AUTOSTART
PROFILE AUTOSTART field (PROFILE_AUTOSTART subsystem parameter)

The PROFILE_AUTOSTART subsystem parameter specifies whether START PROFILE command processing is automatically initiated as part of Db2 start processing.

Acceptable values: NO, YES
Default: NO
Online changeable: no
Update: option 24 on panel DSNTIPB
DSNZP: DSN6SYSP PROFILE_AUTOSTART

NO Profile remain inactive after Db2 is started until a START PROFILE command is issued. NO is the default value.

YES

A START PROFILE command is issued as part of Db2 start processing. The rules specified in profiles become effective immediately at the end of the start process.

This value is ignored for START DB2 commands that specify ACCESS(MAINT) or LIGHT(YES).

Related tasks:

- Starting and stopping profiles (Db2 Performance)
- Using profiles to monitor and optimize Db2 for z/OS subsystems (Db2 Performance)

Related reference:

- START PROFILE (Db2) (Db2 Commands)
- Profile tables (Db2 Performance)

DSNTIPO3: Default startup modules panel

The DSNTIPO3 panel enables you to specify the default member names of certain modules that are to be loaded at Db2 startup. These modules are the subsystem parameter module, the application defaults module, and the exit modules.
**PARAMETER MODULE field**

The value of the PARAMETER MODULE field specifies the member name of the load module that is to be used for Db2 subsystem parameters.

- **Acceptable values:** 1 - 8 characters
- **Default:** DSNZP
- **Update:** option 25 on panel DSNTIPB
- **DSNZPxxx:** none

The parameter module typically resides in library `prefix.SDSNEXIT`. To avoid conflict with members of `prefix.SDSNLOAD`, use DSNZPxxx, where `xxx` is any set of three alphanumeric characters. Db2 puts this name in the startup JCL procedure in SYS1.PROCLIB, but you can override this value by using the START DB2 command.

**APPL DEFAULTS field**

The value of the APPL DEFAULTS field specifies the member name of the load module that is to be used for Db2 application default values.

- **Acceptable values:** for installation: 1 - 8 characters; for migration: DSNHDECP
- **Default:** DSNHDECP
- **Update:** option 25 on panel DSNTIPB
- **DSNHDECP:** none

This module typically resides in library `prefix.SDSNEXIT`. The application default values that are specified within this module are defined by panels DSNTPK, DSNTPM, DSNTIPF, and DSNTIP4. The value that is entered in this field will be assigned to the DECP parameter of the ssmnMSTR startup JCL procedure in SYS1.PROCLIB. The DECP parameter specifies the name of the applications defaults module that is to be loaded by default when Db2 is started. You can override this value by using the DECP option of the START DB2 command. If you use a name that starts with DSN, use DSNHDExx, where `xx` is any set of two alphanumeric characters. This naming convention avoids conflict with members of `prefix.SDSNLOAD`. 

---

Figure 29. Default startup modules panel: DSNTIPO3
If you specify a name other than DSNHDECP, the DSNTIJUZ jobs creates a module with that name, with an alias of DSNHDECP. If alias DSNHDECP already exists, it is replaced. If a module that is named DSNHDECP already exists, the link-edit request fails. In this case, you must delete the DSNHDECP module or remove the ALIAS directive for DSNHDECP from the link-edit in job step DSNTIZQ of DSNTIJUZ.

This parameter cannot be changed online.

**ACCESS CONTROL field (ACCESS_CNTL_MODULE subsystem parameter)**
The value of this field specifies the member name of the load module that is to be used for the Db2 access control exit routine.

Acceptable values: for installation: 1 - 8 characters; for migration: DSNX@XAC
Default: DSNX@XAC
Update: option 25 on panel DSNTIPB
DSNZP: DSN6SYSP ACCESS_CNTL_MODULE

This module typically resides in library prefix.SDSNEXIT. If you specify a name that starts with DSN, use DSNX@xxx, where xxx is any unused set of three alphanumeric characters. This naming convention avoids conflict with members of prefix.SDSNLOAD.

If you specify a name other than DSNX@XAC, the DSNTIJEX job creates a module with that name, with an alias of DSNX@XAC. If alias DSNX@XAC already exists, it is replaced. If a module that is named DSNX@XAC already exists, the link-edit request fails. In this case, you must delete the DSNX@XAC module or remove the ALIAS directive for DSNX@XAC from the link-edit in job step JEX0003 of DSNTIJEX.

This parameter cannot be changed online.

**IDENTIFY/AUTH field (IDAUTH_MODULE subsystem parameter)**
The value of this field specifies the member name of the load module that is to be used for the Db2 connection authorization exit routine.

Acceptable values: for installation: 1 - 8 characters; for migration: DSN3@ATH
Default: DSN3@ATH
Update: option 25 on panel DSNTIPB
DSNZP: DSN6SYSP IDAUTH_MODULE

This module typically resides in library prefix.SDSNEXIT. If you specify a name that starts with DSN, use DSN3@xxx, where xxx is any unused set of three alphanumeric characters. This naming convention avoids conflict with members of prefix.SDSNLOAD.

If you specify a name other than DSN3@ATH, the DSNTIJEX job creates a module with that name, with an alias of DSN3@ATH. If alias DSN3@ATH already exists, it is replaced. If a module that is named DSN3@ATH already exists, the link-edit request fails. In this case, you must delete the DSN3@ATH module or remove the ALIAS directive for DSN3@ATH from the link-edit in job step JEX0002 of DSNTIJEX.

This parameter cannot be changed online.
**SIGNON field (SIGNON_MODULE subsystem parameter)**

The value of this field specifies the member name of the load module that is to be used for the Db2 sign-on exit routine.

Acceptable values: for installation: 1 - 8 characters; for migration: DSN3@SGN
Default: DSN3@SGN
Update: option 25 on panel DSNTIPB
DSNZPxxx: DSN6SYSP SIGNON_MODULE

This module typically resides in library prefix.SDSNEXIT. If you specify a name that starts with DSN, use DSN3@xxx, where xxx is any unused set of three alphanumeric characters. This naming convention avoids conflict with members of prefix.SDSNLOAD.

If you specify a name other than DSN3@SGN, the DSNTIJEX job creates a module with that name, with an alias of DSN3@SGN. If alias DSN3@SGN already exists, it is replaced. If a module that is named DSN3@SGN already exists, the link-edit request fails. In this case, you must delete the DSN3@SGN module or remove the ALIAS directive for DSN3@SGN from the link-edit in job step JEX0001 of DSNTIJEX.

If you run the installation CLIST in MIGRATE mode, this field is always set to DSN3@SGN and cannot be modified.

This parameter cannot be changed online.

**Related concepts:**
- Connection routines and sign-on routines (Managing Security)

**Related tasks:**
- Migration step 16: Define user authorization exit routines: DSNTIJEX (optional)

---

**DSNTIPO4: Resource Limit Facility panel**

The DSNTIPO4 panel enables you to specify options for the resource limit facility.

---

**Figure 30. Default startup modules panel: DSNTIPO4**

**Related concepts:**
The resource limit facility (Introduction to Db2 for z/OS)

Resource limit facility controls (Db2 Performance)

Related tasks:

Setting limits for system resource usage by using the resource limit facility (Db2 Performance)

RLF AUTO START field (RLF subsystem parameter)

The RLF subsystem parameter specifies whether the resource limit facility (governor) is to automatically start each time Db2 is started.

Acceptable values: YES, NO
Default: NO
Update: option 26 on panel DSNTIPB
DSNZPxxx: DSN6SYS PRLF

Related concepts:

Resource limit facility controls (Db2 Performance)

RLF SCOPE field (RLFENABLE subsystem parameter)

The RLFENABLE subsystem parameter specifies whether the resource limit facility limits resources that are used by dynamic SQL statements only, static SQL statement only, or all SQL statements.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: DYNAMIC, STATIC, or ALL
Default: DYNAMIC
Data sharing scope: It is recommend that all members use the same setting.
Update: option 26 on DSNTIPB
DSNZPxxx: DSN6SYS PRLFENABLE

DYNAMIC
Resource limits apply only to dynamic SQL statements.

STATIC
Resource limits apply only to static SQL statements.

ALL
Resource limits apply to dynamic and static SQL statements.

Related concepts:

Resource limit facility controls (Db2 Performance)

Related tasks:

Setting limits for system resource usage by using the resource limit facility (Db2 Performance)

Related reference:

RLF SCOPE field (RLFENABLE subsystem parameter)

RLST NAME SUFFIX field (RLFTBL subsystem parameter)

The RLFTBL subsystem parameter specifies the suffix that is to be used for the default resource limit specification table (RLST). The default RLST is used when the resource limit facility (governor) is automatically started or when the governor is started without a specified suffix.
Acceptable values: any 2 alphanumeric characters; national characters are not allowed
Default: 01
Update: option 2624 on panel DSNTIPB
DSNZPxxx: DSN6SYSP RLFTBL

**DYNAMIC SQL field (RLFERR subsystem parameter)**

The RLFERR subsystem parameter specifies the action that Db2 takes for dynamic SQL statements that originate from the local server when it cannot find an applicable row in the resource limit specification table.

An applicable row is one that applies to the authorization ID, collection ID, package name, and name of the logical unit of work of the query user.

Acceptable values: NOLIMIT, NORUN, 1 to 5000000
Default: NOLIMIT
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6SYSP RLFERR

**NOLIMIT**

Allows all dynamic SQL statements to run without limit.

**NORUN**

Terminates all dynamic SQL statements immediately with an SQL error code.

**1 to 5000000**

Specifies the number of service units to use as the default resource limit. If the limit is exceeded, the SQL statement is terminated.

**Related tasks:**

[Setting limits for system resource usage by using the resource limit facility (Db2 Performance)]

**Related reference:**

[DSNRRLSTxx resource limit tables (Db2 Performance)]

**STATIC SQL field (RLFERRSTC subsystem parameter)**

The RLFERRSTC subsystem parameter specifies the action that Db2 takes when a problem occurs with the resource limit specification table during execution of a local, static SQL statement.

The situations that this subsystem parameter applies to are:

- An error occurs that prevents the resource limit facility from accessing the resource limit specification table.
- Db2 cannot find an applicable row in the resource limit specification table.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: NOLIMIT, NORUN, 1 to 5000000
Default: NOLIMIT
Update: option 26 on DSNTIPB
DSNZPxxx: DSN6SYSP RLFERRSTC
NOLIMIT
No limit is applied to any static SQL statement that does not correspond to a row in the resource limit table.

NORUN
Any static SQL statement that does not correspond to a row in the resource limit table is prevented from running.

1 to 5000000
Specifies the number of service units to use as the default resource limit. If the limit is exceeded, the SQL statement is terminated.

Related concepts:
Resource limit facility controls (Db2 Performance)

Related tasks:
Setting limits for system resource usage by using the resource limit facility (Db2 Performance)

REMOTE DYNAMIC SQL field (RLFERRD subsystem parameter)
The RLFERRD subsystem parameter specifies the default action that Db2 takes for dynamic SQL statements from remote locations when a problem occurs with the resource limit specification table.

The situations that this subsystem parameter applies to are:
- An error occurs that prevents the resource limit facility from accessing the resource limit specification table.
- Db2 cannot find an applicable row in the resource limit specification table.

An applicable row is one that applies to the authorization ID, collection ID, or package name, and name of the logical unit of work of the query user. This parameter applies to the distributed data facility.

Acceptable values: NOLIMIT, NORUN, 1 to 5000000
Default: NOLIMIT
Update: option 26 on panel DSNTIPB
DSNZP:xxx: DSN6FAC RLFERRD

NOLIMIT
Allows all dynamic SQL statements to run without limit.

NORUN
Terminates all dynamic SQL statements immediately with an SQL error code.

1 - 5000000
Specifies the number of service units to use as the default resource limit. If the limit is exceeded, the SQL statement is terminated.

Related concepts:
Resource limit facility controls (Db2 Performance)

Related reference:
DSNRLSTxx resource limit tables (Db2 Performance)

REMOTE STATIC SQL field (RLFERRDSTC subsystem parameter)
The RLFERRDSTC subsystem parameter specifies the action that Db2 takes for static SQL statements from remote locations when a problem occurs with the resource limit specification table.

194 Installation and Migration Guide
The situations that this subsystem parameter applies to are:

- An error occurs that prevents the resource limit facility from accessing the resource limit specification table.
- Db2 cannot find an applicable row in the resource limit specification table.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

An applicable row is one that applies to the authorization ID, collection ID, or package name, and name of the logical unit of work of the query user.

Acceptable values: NOLIMIT, NORUN, 1 - 5000000
Default: NOLIMIT
Update: option 26 on DSNTIPB
DSNZPxxx: DSN6FAC RLFERRDSTC

**NOLIMIT**
Allows all static SQL statements from remote locations to run without limit.

**NORUN**
Terminates all static SQL statements from remote locations immediately with an SQL error code.

**1 - 5000000**
Specifies the number of service units to use as the default resource limit. If the limit is exceeded, the SQL statement is terminated.

### DSNTIPF: Application programming defaults panel 1

The entries on the DSNTIPF panel and the DSNTIP4 panel define application programming default values. These values are used as default values by the program preparation panels, the program preparation CLIST (DSNH), and the precompiler. These values can also be used as default values by other programs, such as QMF.

**Migrating or updating the parameters:** If you alter parameter values for which a change during migration or update is “not recommended,” this change can invalidate the syntax of existing SQL statements or affect the way that application programs run. Update is allowed, but must be handled with caution.

Most of the values that are set here and on the DSNTIP4 panel are contained in the application defaults load module, dsnhdecp. The load module is located in library prefix.SDSNEXIT, which can be loaded and accessed by application programs.

When modifying the application defaults load module, do so only by changing and running the installation CLIST.

**Important:** You should always use the CLIST to modify installation parameters. Do not directly modify the data in the application defaults load module. If you modify any installation parameters by changing job DSNTIJUZ directly, these values are not recorded for later updates, new installations, or migrations. In addition, these values are not checked for validity. If you do not use the CLIST to modify these parameters, Db2 may not start.

Many of the fields on this panel involve the selection of coded character set identifiers (CCSIDs). Here is some information that can help you choose values for these fields:
If you choose YES for the MIXED DATA field, you must specify a mixed data CCSID from Table 146 on page 949 or Table 147 on page 950. An error occurs if you do not specify a CCSID or if the CCSID you specify is not listed in the table.

If you specify an incorrect CCSID, data can become corrupted. For example, assume that the coded character set used at your site is 37, but you specify 500 as the system CCSID. If Db2 receives data with a CCSID of 500, the data can become corrupted because character conversion does not occur. Conversely, if Db2 receives data with a CCSID other than 500 and a conversion is made from that CCSID to 500, the data can become corrupted because character conversion occurs.

**Recommendation:** Never change CCSIDs on an existing Db2 system without specific guidance from IBM Support.

If you need to convert to a CCSID that supports the euro symbol, you can correct it by altering the CCSID field for your default encoding scheme.

During code conversion, Db2 first looks in the SYSSTRINGS table to see if a conversion is defined. If Db2 finds a conversion, it is used. If Db2 does not find a conversion, it uses z/OS Unicode Services. In some cases, z/OS Unicode Services is used instead of the value in SYSSTRINGS. If the conversion is not available, an error occurs.

Converting statements to Unicode for parsing depends on having the correct input CCSID specified. The system CCSIDs must be set up correctly at installation time.

During connect processing, a requester and server provide default CCSIDs for character data sent on the connection.

The Db2 requester uses the application encoding scheme for its default CCSIDs. If an application provides character data that is not in the CCSID that the application encoding scheme identifies, Db2 overrides the default CCSID by tagging each field with the actual CCSID prior to sending to the server. For applications that use the Unicode encoding scheme, the Db2 requester overrides the application encoding scheme CCSIDs with the system EBCDIC CCSIDs and converts the Unicode data to EBCDIC if Db2 determines that the server does not support Unicode character data. If the server cannot accept character data in these CCSIDs, connect fails with a -332 SQLCODE.

The Db2 server uses the system default encoding scheme to determine the default CCSID values for character data that is to be returned to the requester. For servers using the encoding scheme of Unicode, the Db2 server overrides the Unicode encoding scheme CCSIDs with the system EBCDIC CCSIDs and converts the Unicode data to EBCDIC if Db2 determines that the requestor does not support Unicode character data. If the requester cannot accept character data in these CCSIDs, connect fails with a -332 SQLCODE.
LANGUAGE DEFAULT field (DEFLANG DECP value)
The DEFLANG DECP value identifies the default programming language that is to be used for your site.

Acceptable values: ASM, C, CPP, FORTRAN, IBMCOB, PLI
Default: IBMCOB
Update: not recommended

dsnhdecp:: DEFLANG

ASM
Specifies High Level Assembler/MVS as the default programming language.

C
Specifies C as the default programming language. If you specify C, you can fold SQL identifiers to uppercase. However, this is not a default from any installation panel.

CPP
Specifies C++ as the default programming language. If you specify CPP, you can fold SQL identifiers to uppercase. However, this is not a default from any installation panel.

FORTRAN
Specifies Fortran as the default programming language.

IBMCOB
Specifies Enterprise COBOL for z/OS as the default programming language.

PLI
Specifies PL/I as the default programming language.

Related concepts:
Precompiler option list format (Db2 Application programming and SQL)

DECIMAL POINT IS field (DECIMAL DECP value)
The DECIMAL DECP value determines whether the decimal point for numbers is to be a comma (,) or a period (.).

Acceptable values: . (period) or , (comma)
Default: . (period)
Update: recommended only to recover an error
dsnhdecp: DECIMAL

For example, some nations customarily signify the number “one and one-half,” as 1.5; other nations use 1,5 for the same value.

This parameter is used in the following cases:
• For running dynamic SQL statements with DYNAMICRULES:
  – Whether the value of field DECIMAL POINT IS is COMMA or PERIOD, Db2 recognizes the value as the decimal point for numbers.
• For binding, defining, or invoking dynamic SQL statements with DYNAMICRULES:
  – If the value of field USE FOR DYNAMICRULES is NO, Db2 does not use the value in field DECIMAL POINT IS if you specify the option COMMA or PERIOD when you precompile the application that contains the dynamic SQL statements. Db2 uses the precompiler option to determine the decimal point for numbers.
  – If the value of field USE FOR DYNAMICRULES is YES, and the value of field DECIMAL POINT IS is PERIOD or COMMA, Db2 recognizes the value as the decimal point for numbers.
• For static SQL statements in COBOL programs, DECIMAL POINT IS specifies the default precompiler option (PERIOD or COMMA).

This parameter is the default for binds at this Db2 site that are requested by a remote system that does not indicate whether the period or the comma is used to represent a decimal point. In most cases, however, requesting systems give Db2 this information.
Related concepts:
[Decimal point representation (Db2 SQL)]

Related reference:
[DynamicRULES bind option (Db2 Commands)]

STRING DELIMITER field (DELIM DECP value)
The DELIM DECP value determines the string delimiter that is to be used for COBOL.

Acceptable values: DEFAULT, " (quotation mark), ’ (apostrophe)
Default: DEFAULT
Update: not recommended
dsnhdecp: DELIM

This option is effective for all varieties of COBOL.
DEFAULT

The string delimiter is the quotation mark.

" The string delimiter is the quotation mark.

' The string delimiter is the apostrophe.

See the SQL STRING DELIMITER field for a description of how to use this field to get the set of character string delimiters for COBOL and SQL.

SQL STRING DELIMITER field (SQLDELI DECP value)

The SQLDELI DECP value specifies the character that is to be used as the SQL string delimiter to delimit character strings in dynamic SQL.

Acceptable values: DEFAULT, " (quotation mark), ' (apostrophe)
Default: DEFAULT
Update: not recommended
dsnhdecp: SQLDELI

This option is effective for all varieties of COBOL.

The value in this field also determines which character is the escape character for delimited identifiers in dynamic SQL. If you specify an apostrophe in this field, you get a quotation mark for your SQL escape character. If you specify a quotation mark in this field, you get an apostrophe for your SQL escape character.

For SQL statements that are embedded in COBOL programs, COBOL precompiler options specify which character is the SQL string delimiter and which character is the SQL escape character. If you specify DEFAULT in this field, a quotation mark is passed to the precompiler as the default SQL string delimiter.

Some applications might require a particular value for the SQL STRING DELIMITER. Determine the required values for those applications before installing Db2.

The following table shows the different combinations of character string delimiters that result from specifying different values in the STRING DELIMITER field and the SQL STRING DELIMITER field.

Table 26. Effect of STRING DELIMITER field and SQL STRING DELIMITER field on SQL and COBOL string delimiters

<table>
<thead>
<tr>
<th>When you want this combination of character string delimiters...</th>
<th>Specify this in STRING DELIMITER field</th>
<th>Specify this in SQL STRING DELIMITER field</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBOL</td>
<td>Dynamic SQL</td>
<td>Embedded SQL</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
<td>&quot; &quot;</td>
</tr>
</tbody>
</table>

The values that you specify in the STRING DELIMITER field and the SQL STRING DELIMITER field are also used by the program preparation panels, the DSNH CLIST, and the precompiler. The following table shows why you might specify different combinations of values in these fields.

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST 199
Table 27. Effect of the STRING DELIMITER field and the SQL STRING DELIMITER field on precompiler options

<table>
<thead>
<tr>
<th>Purpose</th>
<th>STRING DELIMITER field</th>
<th>SQL STRING DELIMITER field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force APOST default (even in COBOL) and provide a default similar to APOST in Db2 Server for VSE &amp; VM.</td>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>Change dynamic query string delimiter to the quotation mark. This is helpful if you use COBOL with the QUOTE option. It allows queries to be tested with dynamic SQL and moved into the program more easily.</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>Enables compatibility with the Db2 Server for VSE &amp; VM option.</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

**DIST SQL STR DELIMTR field (DSQLDELI DECP value)**

The DSQLDELI DECP value specifies whether an apostrophe or quotation mark is to be used as the SQL string delimiter for bind operations at this Db2 site. This delimiter is to be used when the requester does not give Db2 that information.

In most cases, requesters tell Db2 whether the apostrophe or the quotation mark is to be used as the SQL string delimiter.

Acceptable values: ' (apostrophe) or " (quotation mark)
Default: ' (apostrophe)
Update: not recommended
dsnhdecp: DSQLDELI

**MIXED DATA field (MIXED DECP value)**

The value of the MIXED DECP field indicates how the EBCDIC CCSID and ASCII CCSID fields are to be interpreted by Db2.

The MIXED DATA option has no effect on a Unicode string, including a Unicode column in an EBCDIC table. Regardless of the setting for MIXED DATA, UNICODE UTF-8 data is considered mixed data and is processed according to the rules for mixed data.

Acceptable values: YES, NO
Default: NO
Update: not recommended
dsnhdecp: MIXED

**YES**

The CCSID that is specified in the EBCDIC CCSID and ASCII CCSID field must be the mixed CCSID for the encoding scheme. From this, Db2 determines the associated SBCS and DBCS CCSIDs for the encoding scheme. MIXED DATA YES allows EBCDIC and ASCII mixed-character data and graphic data to be defined.

**Important:** MIXED DATA applies to both the EBCDIC CCSID and ASCII CCSID. If you choose MIXED DATA YES, you must select mixed CCSIDs for EBCDIC and ASCII.

**NO**

The CCSID specified in the EBCDIC CCSID or ASCII CCSID field must be the
CCSID for the encoding scheme. MIXED DATA NO does not allow for EBCDIC or ASCII mixed character data or graphic data to be defined.

For EBCDIC data, the value specifies whether the code points X'0E' and X'0F' have special meaning as the shift-out and shift-in controls for character strings that include double-byte characters.

- NO indicates that these code points have no special meaning. Therefore, all character strings are single-byte character set (SBCS) data.
- YES indicates that these code points have the special meaning described above. Therefore, character strings can be either SBCS or MIXED data.

Related reference:

**EBCDIC CCSID field (SCCSID, MCCSID, and GCCSID DECP values)**

The value of the EBCDIC CCSID field specifies the default CCSID for EBCDIC-encoded character data that is stored in your Db2 subsystem or data sharing system.

Db2 uses this value to perform conversion of character data that is received from external sources including other database management systems. Choose this value carefully to avoid loss of data integrity. The values that you choose for EBCDIC CCSID and ASCII CCSID are closely related.

Acceptable values: 1 - 65533
Default: (none)
Update: not recommended; data integrity may be compromised

**dsnhdecp:** SCCSID (single-byte), MCCSID (mixed), GCCSID (graphic)

If you specify NO for the MIXED DATA field, the MCCSID and GCCSID values are 65534. If you specify YES for the MIXED DATA field, ensure that you use the correct single-byte CCSID and MCCSID.

**Recommendation:** Use this parameter to specify an MCCSID. By doing so, you also receive system CCSIDs for your SBCS and GRAPHIC data. If you edit DSNHDECP directly, you will not receive the system CCSIDs for your SBCS and GRAPHIC data.

Conversions are determined in the following order:
1. SYSIBM.SYSSTRINGS.
2. z/OS Unicode conversion services.

*Considerations for mixed data:*

- If the MIXED DATA field is set to YES, you must specify a MCCSID from EBCDIC and ASCII support. An error occurs if you do not specify a CCSID or if the CCSID you specify is not listed in the table.
- If you specify 930, 1390, or 5026, Katakana characters are allowed in ordinary identifiers, and letters are not changed to uppercase.

If you specify a CCSID that is recognized by Db2 but is inappropriate for your site, data might be corrupted. For example, assume that the coded character set at your site is 37, but you specify 500 as the system CCSID. If Db2 receives data with a CCSID of 37, the data might be corrupted because character conversion does not...
occur. Conversely, if Db2 receives data with a CCSID other than 500 and a conversion is made from that CCSID to 500, the data may be corrupted because character conversion does occur.

Altering CCSIDs can be very disruptive to a system. Converting to a CCSID that supports the euro symbol is potentially less disruptive because specific pre-euro CCSIDs map to specific CCSIDs for the euro. Converting to a different CCSID for other reasons, particularly when a Db2 system has been operating with the wrong CCSID, could render data unusable and unrecoverable.

**Recommendation**: Never change CCSIDs on an existing Db2 system without specific guidance from IBM Software Support.

**Related concepts**: Euro symbol support

**Related reference**: EBCDIC and ASCII support

**Related information**: Character Data Representation Architecture Reference

**ASCII CCSID field (ASCCSID, AMCCSID, AGCCSID DECP values)**

The ASCII CCSID field identifies the default coded character set identifier (CCSID) for ASCII-encoded character data that is to be stored in your Db2 subsystem or data sharing system.

Db2 uses the value of this field to convert character data that is received from external sources, including other database management systems. You must specify a value for this field, even if you do not have or plan to create ASCII-encoded objects. Choose this value carefully to prevent loss of data integrity.

**Acceptable values**: 1 - 65533

**Default**: (none)

**Update**: not recommended; data integrity may be compromised

**dsnhdecp**: ASCCSID (single-byte), AMCCSID (mixed), AGCCSID (graphic)

**Recommendation**: Never change CCSIDs on an existing Db2 system without specific guidance from IBM Software Support.

**Related reference**: EBCDIC and ASCII support

**EBCDIC CCSID field (SCCSID, MCCSID, and GCCSID DECP values)**

**UNICODE CCSID field (USCCSID, UMCCSID, and UGCCSID DECP values)**

The value of the UNICODE CCSID field specifies the CCSID that is to be used for Unicode data.

**Acceptable values**: 1208

**Default**: 1208

**Update**: not recommended; data integrity may be compromised

**dsnhdecp**: USCCSID (367 for single-byte), UMCCSID (1208 for mixed), UGCCSID (1200 for graphic)
Accept the default CCSID for Unicode. Db2 currently allows specification of only CCSID 1208 for this value. Db2 automatically chooses the CCSIDs for double-byte and single-byte data. Do not change CCSID values after they have been specified. SQL results might be unpredictable if you do not accept the default.

**DEF ENCODING SCHEME field (ENSCHEME DECP value)**

The ENSCHEME DECP value specifies the default format in which data is to be stored in Db2.

Acceptable values: EBCDIC, ASCII, UNICODE

Default: EBCDIC

Update: not recommended

dsnhdecp: ENSCHEME

If you set DEF ENCODING SCHEME to ASCII or EBCDIC and MIXED DATA is set to YES, specify a mixed CCSID.

The DDL uses the default encoding scheme in the following cases:

- CREATE DATABASE
- CREATE DISTINCT TYPE
- CREATE FUNCTION
- CREATE GLOBAL TEMPORARY TABLE
- DECLARE GLOBAL TEMPORARY TABLE
- CREATE TABLESPACE (in DSNDB04 database)

**APPLICATION ENCODING field (APPENSCH DECP value)**

The APPENSCH DECP value determines the system's default application encoding scheme, which affects how Db2 interprets data coming into Db2.

For example, if your default application encoding scheme is set to 37, and your EBCDIC coded character is set to 500, Db2 converts all data coming into the system to 500 from 37 before using it. This includes, but is not limited to, SQL statement text and host variables.

Acceptable values: ASCII, EBCDIC, UNICODE, or CCSID

Default: EBCDIC

Update: not recommended

dsnhdecp: APPENSCH

The following statements set the value of the host variable and do not require the package to be bound into the plan:

```
SET CURRENT PACKAGE SET = :HV ,
SET :HV = CURRENT PACKAGE SET ,
SET :HV = CURRENT PACKAGE PATH ,
SET CURRENT PACKAGE PATH = :HV
```

The host variable uses the system's default application encoding scheme, even when the application is bound with the ENCODING(EBCDIC/UNICODE) bind option.

The default value, EBCDIC, causes Db2 to retain the behavior of previous versions of Db2. (Assume that all data is in the EBCDIC system CCSID.)
**LOCALE LC_CTYPE field (LC_TYPE DECP value)**

The LC_TYPE DECP value specifies the system LOCALE LC_CTYPE. A locale is the part of your system environment that depends on language and cultural conventions. An LC_CTYPE is a subset of a locale that applies to character functions.

Acceptable values: A valid locale of 0 to 50 characters.
Default: Blank
Update: not recommended
dsnhdecp: LC_TYPE

The UPPER, LOWER, and TRANSLATE scalar functions use the CURRENT LOCALE LC_CTYPE system default or special register. The results of these functions can vary, depending on the setting of the locale.

**Recommendation:** Use the default value for LOCALE LC_CTYPE unless you need to execute the UPPER, LOWER, or TRANSLATE functions for data that must be interpreted by using the rules provided by specific locales. For example, specify En_US for English in the United States or Fr_CA for French in Canada.

**Related reference:**

**DECFLOAT ROUNDING MODE field (DEF_DECFLOAT_ROUND_MODE DECP value)**

The DEF_DECFLOAT_ROUND_MODE DECP value specifies the system default action that is to be used for rounding decimal floating point values.

Acceptable values: ROUND_CEILING, ROUND_DOWN, ROUND_FLOOR, ROUND_HALF_DOWN, ROUND_HALF_EVEN, ROUND_HALF_UP, ROUND_UP
Default: ROUND_HALF_EVEN
Update: not recommended
dsnhdecp: DEF_DECFLOAT_ROUND_MODE

**ROUND_CEILING**
Rounds towards +infinity. If all of the discarded digits are zero or if the sign is negative, Db2 removes the discarded digits and the rest of the result remains the same. In all other cases, Db2 increments the result coefficient by 1.

**ROUND_DOWN**
Rounds towards 0. Db2 ignores the discarded digits.

**ROUND_FLOOR**
Rounds towards -infinity. If all of the discarded digits are zero or if the sign is positive, Db2 removes the discarded digits and the rest of the result remains the same. In all other cases, Db2 increments the result coefficient by 1.

**ROUND_HALF_DOWN**
Rounds to the nearest number. If the discarded digits are greater than 0.5 of the value of a one in the next left position, Db2 increments the result coefficient by 1. If the discarded digits are equal or less than 0.5, Db2 ignores the discarded digits.

**ROUND_HALF_EVEN**
Rounds to the nearest number. If the discarded digits are greater than 0.5, the
value of a one in the next left position, Db2 increments the result coefficient by 1. If the discarded digits are less than 0.5, Db2 ignores the discarded digits. If the discarded digits are 0.5 and the rightmost digit is even, Db2 does not alter the result coefficient. If the discarded digits are 0.5 and the rightmost digit is odd, Db2 increments the result coefficient by 1 to make it an even digit.

**ROUND_HALF_UP**

Rounds to the nearest number. If the discarded digits are greater than or equal to 0.5 of the value of a one in the next left position, Db2 increments the result coefficient by 1. In all other cases, Db2 ignores the discarded digits.

**ROUND_UP**

Rounds away from 0. If all of the discarded digits are zero, Db2 removes the discarded digits and the result of the result remains the same. In all other cases, Db2 increments the result coefficient by 1.

Related reference:
- [BIND and REBIND options for packages, plans, and services (Db2 Commands)](https://www.ibm.com/docs/en/db2?topic=commands-binde-option)

**DSNTP4: Application programming defaults panel 2**

The DSNTP4 panel is a continuation of the DSNTPF panel, and it is used to set application programming default values. The values that you specify on this panel are used as default values by the program preparation panels, the program preparation CLIST (DSNH), and the precompiler.

The values that you specify on this panel can also be used as default values by other programs, such as QMF.

<table>
<thead>
<tr>
<th>Enter data below:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MINIMUM DIVIDE SCALE -&gt; NO NO or YES for a minimum of 3 digits to right of decimal after division</td>
</tr>
<tr>
<td>2 DECIMAL ARITHMETIC -&gt; DECIS DECIS, DECISI, ISI, ISI1 or DPP.S</td>
</tr>
<tr>
<td>3 USE FOR DYNAMICRULES -&gt; YES YES or NO</td>
</tr>
<tr>
<td>4 DESCRIBE FOR STATIC -&gt; YES Allow DESCRIBE for STATIC SQL. NO or YES</td>
</tr>
<tr>
<td>5 DATE FORMAT -&gt; ISO ISO, JIS, USA, EUR, LOCAL</td>
</tr>
<tr>
<td>6 TIME FORMAT -&gt; ISO ISO, JIS, USA, EUR, LOCAL</td>
</tr>
<tr>
<td>7 LOCAL DATE LENGTH -&gt; 0 10-254 or 0 for no exit</td>
</tr>
<tr>
<td>8 LOCAL TIME LENGTH -&gt; 0 8-254 or 0 for no exit</td>
</tr>
<tr>
<td>9 IMPLICIT TIME ZONE -&gt; CURRENT Implicit time zone for TIMESTAMP values (CURRENT, SESSION, or -12:00 to +14:00)</td>
</tr>
<tr>
<td>10 STD SQL LANGUAGE -&gt; NO NO or YES</td>
</tr>
<tr>
<td>11 PAD NUL-TERMINATED -&gt; NO NO or YES</td>
</tr>
</tbody>
</table>

PRESS: ENTER to continue RETURN to exit HELP for more information

*Figure 32. Application programming defaults panel: DSNTP4*
MINIMUM DIVIDE SCALE field (DECDIV3 subsystem parameter)
The DECDIV3 subsystem parameter determines whether to retain at least three digits to the right of the decimal point after any decimal division.

Acceptable values: YES, NO
Default: NO
Update: option 28 on panel DSNTIPB
DSNZPxxx: DSN6SPRM DECDIV3

Certain accounting applications might need this option.

YES
Retains at least three digits to the right of the decimal point after any decimal division.

NO
Does not retain three digits to the right of the decimal point after decimal division. Use NO, the default, to accept the usual rules for decimal division in SQL.

Related concepts:

DECIMAL ARITHMETIC field (DECARTH DECP value)
The DECARTH DECP value specifies the rules that are to be used when both operands in a decimal operation have precisions of 15 or less.

Acceptable values: DEC15, DEC31, 15, 31, or DPP.S
Default: DEC15
Update: not recommended; cannot be changed during migration
dsnhdcecp: DECARTH

DEC15 or 15
Specifies the rules that do not allow a precision greater than 15 digits.

DEC31 or 31
Specifies the rules that allow a precision of up to 31 digits. The rules for DEC31 are always used if either operand has a precision greater than 15. If you chose DEC15 for your previous installation, choosing DEC31 can produce different results for operations on existing data.

DPP.S
“PP” is either 15 or 31 and “S” is a number between 1 and 9. If the form “DPP.S” is used, “PP” represents the precision that is to be used with the rules that are used for DEC15 or DEC31, and “S” represents the minimum divide scale to use for division operations. The separator that is used in the form “DPP.S” can be either the “.” or the “,” character, regardless of the setting of the default decimal point. An error occurs if any other values are specified.

This installation option applies to dynamic SQL by becoming the initial value for the CURRENT PRECISION special register, and it provides the default for the DEC precompiler option. DEC15 is sufficient for most sites. Do not choose DEC31 unless...
you are certain that you need the extra precision. If you use DEC31, you are more likely to get a bind error, particularly in division operations.

Related concepts:

- Arithmetic with two decimal operands (Db2 SQL)

**USE FOR DYNAMICRULES field (DYNRULES DECP value)**

The DYNRULES DECP value specifies whether Db2 is to use the application programming default values that are specified on this panel. The other option is to use the values of the Db2 precompiler options for dynamic SQL statements that are bound by using DYNAMICRULES bind, define, or invoke behavior.

Acceptable values: YES or NO
Default: YES
Update: not recommended

dsnhdecp: DYNRULES

**YES**

Db2 uses the application programming defaults for these fields regardless of the DYNAMICRULES option:

- DECIMAL POINT IS
- STRING DELIMITER
- SQL STRING DELIMITER
- MIXED DATA
- DECIMAL ARITHMETIC

**NO**

Db2 uses the Db2 precompiler values for dynamic SQL statements in plans or packages that are bound by using the DYNAMICRULES bind, define, or invoke behavior.

**DESCRIBE FOR STATIC field (DESCSTAT subsystem parameter)**

The DESCSTAT subsystem parameter controls whether Db2 is to build a DESCRIBE SQL descriptor area (SQLDA) when binding static SQL statements. DESCRIBE SQLDAs allow DESCRIBE requests for static SQL statements to execute successfully.

The DESCSTAT subsystem parameter provides the default value for the DESCSTAT BIND or REBIND option. The DESCSTAT BIND or REBIND option value always overrides the DESCSTAT subsystem parameter value.

Normally, a DESCRIBE request cannot be issued against a static SQL statement, with the following exceptions:

- In a distributed environment, where Db2 for z/OS is the server, and the requester supports extended dynamic SQL. In this scenario, a DESCRIBE request that is executed on an SQL statement in the extended dynamic package appears to Db2 as a DESCRIBE on a static SQL statement in the Db2 package.
- When an application uses a stored procedure result set, and the application must allocate a cursor for that result set. The application can describe that cursor by using a DESCRIBE CURSOR statement. The SQL statement that is actually described is the one for which the cursor is declared in the stored procedure. If that statement is static, this requires that a static SQL statement must be described.

Acceptable values: NO or YES
Default: YES
Db2 does not generate a DESCRIBE SQLDA at bind time for static SQL statements. If a DESCRIBE request is received at execution time, Db2 generates an error. However, if the describe request comes from a DESCRIBE CURSOR statement, Db2 satisfies the request but is able to provide only data type and length information. Column names are not provided.

**YES**

Db2 generates a DESCRIBE SQLDA at bind time so that DESCRIBE requests for static SQL can be satisfied during execution. You must rebind the package after this value has been set to YES. Specifying YES increases the size of some packages because the DESCRIBE SQLDA is now stored with each statically bound SQL SELECT statement.

YES is the default value.

If your Db2 for z/OS subsystem or Db2 for Linux, UNIX, and Windows systems use the IBM Data Server Driver for JDBC and SQLJ, or if your Db2 for Linux, UNIX, and Windows systems use the CLI driver, you must set the DESCSTAT parameter to YES.

**Related reference:**

- [BIND and REBIND options for packages, plans, and services (Db2 Commands)](https://www.ibm.com/support/docview.ws/dockey/1547118436)

**DATE FORMAT field (DATE DECP value)**

The DATE DECP value controls the default format that is to be used to represent dates in output.

Acceptable values: ISO, USA, EUR, JIS, LOCAL

Default: ISO

Update: not recommended

dsnhdcecp: DATE

The date formats that are associated with the acceptable values for this field are shown in the table below.

Table 28. Date formats

<table>
<thead>
<tr>
<th>Format name</th>
<th>Abbreviation</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>International Standards Organization</td>
<td>ISO</td>
<td>yyyy-mm-dd</td>
<td>2003-12-23</td>
</tr>
<tr>
<td>IBM USA standard</td>
<td>USA</td>
<td>mm/dd/yyyy</td>
<td>12/23/2003</td>
</tr>
<tr>
<td>IBM European standard</td>
<td>EUR</td>
<td>dd.mm.yyyy</td>
<td>23.12.2003</td>
</tr>
<tr>
<td>Japanese Industrial Standard Christian Era</td>
<td>JIS</td>
<td>yyyy-mm-dd</td>
<td>2003-12-23</td>
</tr>
<tr>
<td>Locally defined (by an installation exit routine)</td>
<td>LOCAL</td>
<td>your choice</td>
<td></td>
</tr>
</tbody>
</table>

Db2 accepts dates in any format as input. Db2 interprets the input date based on the punctuation, and then provides date output in the format that you specify for this parameter. If you use LOCAL, you must provide a date exit routine to perform date formatting.
Related concepts:

[Date and time routines (Db2 Administration Guide)]

**TIME FORMAT field (TIME DECP value)**

The value of the TIME FORMAT field specifies which format is to be used as a default output to represent times.

Acceptable values: ISO, USA, EUR, JIS, LOCAL

Default: ISO

Update: not recommended

dsnhexcp: TIME

**Table 29. Time formats**

<table>
<thead>
<tr>
<th>Format name</th>
<th>Abbreviation</th>
<th>Time format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>International Standards Organization¹</td>
<td>ISO¹</td>
<td>hh.mm.ss</td>
<td>13.30.05</td>
</tr>
<tr>
<td>IBM USA standard</td>
<td>USA</td>
<td>hh:mm AM or PM</td>
<td>1:30 PM</td>
</tr>
<tr>
<td>IBM European standard</td>
<td>EUR</td>
<td>hh.mm.ss</td>
<td>13.30.05</td>
</tr>
<tr>
<td>Japanese Industrial Standard Christian Era</td>
<td>JIS</td>
<td>hh:mm:ss</td>
<td>13:30:05</td>
</tr>
<tr>
<td>Installation-defined</td>
<td>LOCAL</td>
<td>Any installation-defined form</td>
<td>—</td>
</tr>
</tbody>
</table>

*Note:* 1. This is an earlier version of the ISO format. JIS can be used to get the current ISO format.

In the USA format:

- The minutes can be omitted, thereby specifying 00 minutes. For example, 1 PM is equivalent to 1:00 PM.
- The letters A, M, and P can be lowercase.
- A single blank must precede the AM or PM.
- The hour must not be greater than 12 and cannot be 0 except for the special case of 00:00 AM.

Using the ISO format of the 24-hour clock, the correspondence between the USA format and the 24-hour clock is as follows:

- 12:01 AM through 12:59 AM correspond to 00.01.00 through 00.59.00.
- 01:00 AM through 11:59 AM correspond to 01.00.00 through 11.59.00.
- 12:00 PM (noon) through 11:59 PM correspond to 12.00.00 through 23.59.00.
- 12:00 AM (midnight) corresponds to 24.00.00.
- 00:00 AM (midnight) corresponds to 00.00.00.

Db2 can accept a time in any format as input. Db2 interprets the input time based on the punctuation and then provides time output in the format that you specify for this parameter. If you use LOCAL, you must provide a time exit routine to perform time formatting.

Related concepts:

[Date and time routines (Db2 Administration Guide)]
LOCAL DATE LENGTH field (DATELEN DECP value)
If you use a locally defined date exit routine, the DATELEN DECP value defines the length of the longest field that is required to hold a date. To use your own default date format, you must also enter LOCAL for the DATE FORMAT field on panel DSNTIP4.

Acceptable values: 0, 10 - 254
Default: 0
Update or migrate: not recommended
dsnhtdecp: DATELEN

Accept the default value of 0 if you want to use one of the IBM-supplied date formats (ISO, JIS, USA, or EUR). This value indicates that no user-defined date format exists in your system.

LOCAL TIME LENGTH field (TIMELEN DECP value)
Accept the default value of 0 if you want to use one of the IBM-supplied time formats (ISO, JIS, USA, or EUR).

Acceptable values: 0, 8 - 254
Default: 0
Update or migrate: not recommended
dsnhtdecp: TIMELEN

This indicates that no user-defined time format exists in your system. If you use a locally defined time exit routine, enter the length of the longest field that is required to hold a time. If you want your own time format to be the default, enter LOCAL for field 2 on this panel.

IMPLICIT TIME ZONE field (IMPLICIT_TIMEZONE DECP value)
The IMPLICIT_TIMEZONE DECP value determines the implicit time zone that is to be used when a time zone is not provided. This parameter applies to Db2 table columns and routing parameters that are declared with TIMESTAMP WITH TIME ZONE data types.

Acceptable values: CURRENT, SESSION, -12:59 to +14:00
Default: CURRENT
Update: not recommended

dsnhtdecp: IMPLICIT_TIMEZONE

1Given the way that the implicit time zone is defined, a query might return different results before and after the IMPLICIT_TIMEZONE setting is changed. Or, if IMPLICIT_TIMEZONE is set to SESSION and an application changes the value of SESSION TIME ZONE through a SET SESSION TIME ZONE statement, a query might also return a different result. These situations apply to both static and dynamic applications.

CURRENT
Uses the setting of the CURRENT TIME ZONE special register as the implicit time zone. This is the default setting.

SESSION
Uses the setting of the SESSION TIME ZONE special register as the implicit time zone.

-12:59 to +14:00
Uses the value of the string as the implicit time zone. The format of the value is th:tm, where th is the time zone hour (-12 to +14), and tm is the time zone
minute (00 - 59). This value is an offset of hours and minutes from the
coordinated universal time (UTC). If the string does not indicate '+' or '-', a '+'
value is assumed

STD SQL LANGUAGE field (STDSQL DECP value)
The STD SQL LANGUAGE field specifies the SQL standard that is to be used in
writing application programs. Specifically, the parameter specifies whether the SQL
language conforms to the portions of the 1992 ANSI SQL standard that are
implemented by Db2.

Acceptable values: YES, NO
Default: NO
Update: not recommended
dsnhdecp: STDSQL

YES
The SQL language that is used in application programs conforms to the
portions of the 1992 ANSI SQL standard that are implemented by Db2.

NO
Programs are written in accordance with the SQL language that is defined by
Db2.

PAD NUL-TERMINATED field (PADNTSTR DECP value)
The PADNTSTR subsystem parameter determines whether output host variables
that are nul-terminated strings are to be padded with blanks and a nul-terminator.

Acceptable values: NO, YES
Default: NO
Update: not recommended
dsnhdecp: PADNTSTR

NO
Nul-terminated output host variables have the nul-terminator placed at the end
of actual data that is returned in the host variable.

YES
Nul-terminated output host variables have the nul-terminator placed at the end
of the string, after the string has been padded with blanks from the end of the
actual data to the length of the output host variable.

DSNTIP8: Performance and optimization panel
The DSNTIP8 panel is a continuation of the DSNTIP4 panel. It is used to set
application programming default values pertaining to performance and
optimization.
Figure 33. Performance and optimization panel: DSNTIP8

CACHE DYNAMIC SQL field (CACHEDYN subsystem parameter)
The CACHEDYN subsystem parameter controls whether prepared, dynamic SQL statements are to be cached for later use by eligible application processes.

Acceptable values: YES, NO
Default: YES
Update: option 29 on panel DSNTIPB
DSNZPxxx: DSN6SPRM CACHEDYN
Security parameter: Yes

YES
Causes prepared, dynamic SQL statements to be cached in the EDM dynamic statement cache. If you specify YES, consider this usage when you calculate your EDM pool size. If you specify YES, you must also specify YES for the USE PROTECTION field on panel DSNTIPP.

NO
Causes prepared, dynamic SQL statements to not be cached.

Note: This is a security-related parameter. When RACF authorization is used and this parameter is set, the changes that are made in RACF profiles are not immediately reflected. This setting might result in a user (cached AUTHID) being able to issue the dynamic statements until the cache is refreshed.

Related tasks:
- Improving dynamic SQL performance (Db2 Performance)

Related reference:
- USE PROTECTION field (AUTH subsystem parameter)

Related information:
- Dynamic Statement Cache (white paper)
CACHE DYN STABILITY field (CACHE_DYN_STABILIZATION subsystem parameter)

The CACHE_DYN_STABILIZATION subsystem parameter specifies whether dynamic SQL statements can be captured for stabilization and whether Db2 uses the captured run time structures to process stabilized dynamic SQL statements that are not present in the dynamic statement cache. Db2 avoids processing full prepares when it uses the captured run time structures.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: BOTH, CAPTURE, LOAD, NONE
Default: BOTH
Data sharing scope: It is recommended that all members use the same setting.
Update: option 29 on panel DSNTIIPB
Online changeable: Yes
DSNZPxxx: DSN6SPRM.CACHEDYN_STABILIZATION

The following table summarizes the effects of the CACHE_DYN_STABILIZATION subsystem parameter on the processing of START DYNQUERYCAPTURE commands and dynamic SQL statements:

Table 30. Effects of CACHE_DYN_STABILIZATION subsystem parameter values

<table>
<thead>
<tr>
<th>CACHE_DYN_STABILIZATION value</th>
<th>Db2 accepts START DYNQUERYCAPTURE commands and captures run time structures for in-progress MONITOR(YES) commands?</th>
<th>Db2 dynamic SQL by uses captured run time structures to avoid full prepares for dynamic SQL?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOTH</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CAPTURE</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LOAD</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>NONE</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**BOTH**
Dynamic SQL statements are captured to the catalog, and Db2 uses captures access paths for stabilized statements.

**CAPTURE**
Dynamic SQL statements can be captured to the catalog, but Db2 does not use any captured access paths.

**LOAD**
Db2 uses captures access paths captured access paths for dynamic SQL statements, but statements cannot be captured for stabilization.

**NONE**
Dynamic statements cannot be captured and Db2 does not use any captured access paths.

Related tasks:
- Stabilizing access paths for dynamic SQL statements (Db2 Performance)

Related reference:
- START DYNQUERYCAPTURE (Db2) (Db2 Commands)
OPTIMIZATION HINTS field (OPTHINTS subsystem parameter)

The OPTHINTS subsystem parameter specifies whether Db2 applies various methods for influencing access path selection when static SQL statements are bound or dynamic SQL statements are prepared.

Db2 applies matching various methods for influencing access path selection for SQL statements only when the value of the OPTHINTS subsystem parameter is set to YES.

Examples of these methods include:
- Statement-level predicate selectivity overrides.
- Statement-level access paths.
- Statement-level optimization parameters.
- PLAN_TABLE access path hints.

Acceptable values: NO, YES
Default: NO
Update: option 29 on panel DSNTPB
DSNZPxxx: DSN6SPRM OPTHINTS

If you accept the default value, the methods are not applied and the statements are optimized normally.

When the value is set to YES, Db2 applies hints in the following situations:
- For rows in a PLAN_TABLE instance that contain a value in the OPTHINT column that matches the value of the OPTHINT bind option or the CURRENT OPTIMIZATION HINT special register.
- For any instance of a statement whose application environment and text matches rows in the SYSIBM.SYSQUERY catalog table.

Related tasks:
- Preparing to influence access paths (Db2 Performance)
- Influencing access path selection (Db2 Performance)
- Managing and preventing access path change (Db2 Performance)

Related reference:
- PLAN_TABLE (Db2 Performance)
- SYSQUERY catalog table (Db2 SQL)
- CURRENT OPTIMIZATION HINT (Db2 SQL)
- BIND and REBIND options for packages, plans, and services (Db2 Commands)

EVALUATE UNCOMMITTED field (EVALUNC subsystem parameter)

The EVALUNC subsystem parameter controls whether predicate evaluation is to be allowed on uncommitted data of other transactions. This parameter applies only to stage 1 predicate processing that uses table access for queries with isolation level RS or CS. Table access includes table space scan, index-to-data access, and RID-list processing.
The EVALUNC subsystem parameter applies to predicates in queries, searched UPDATE statements, and searched DELETE statements.

EVALUNC does not apply to the following cases:
- To predicates in SQL statements that access the Db2 catalog or directory
- To data definition statements, such as DROP TABLESPACE, for which internal processes query catalog or directory tables

Acceptable values: NO, YES
Default: NO
Update: option 29 on panel DSNTIPB
DSNZPxxx: DSN6SPRM EVALUNC

NO  Predicate evaluation occurs only on committed data (or on the application’s own uncommitted changes). NO ensures that all qualifying data is always included in the answer set.

This is the default value.

YES  Predicate evaluation can occur on uncommitted data of other transactions.
With YES, data might be excluded from the answer set. Data that does not satisfy the predicate during evaluation but then, because of undo processing (ROLLBACK or statement failure), reverts to a state that does satisfy the predicate is missing from the answer set. A value of YES enables Db2 to take fewer locks during query processing. The number of avoided locks depends on:
  - The query’s access path
  - The number of evaluated rows that do not satisfy the predicate
  - The number of those rows that are on overflow pages

Recommendation: Specify YES to improve concurrency if your applications can tolerate returned data that might falsely exclude any data that would be included as the result of undo processing (ROLLBACK or statement failure).

Although the option influences whether predicate evaluation can occur on uncommitted data, it does not influence whether uncommitted data is returned to an application. Queries with isolation level RS or CS return only committed data. They never return the uncommitted data of other transactions, even if predicate evaluation occurs. If data satisfies the predicate during evaluation, the data is locked as needed, and the predicate is re-evaluated as needed before the data is returned to the application.

SKIP UNCOMM INSERTS field (SKIPUNCI subsystem parameter)
The SKIPUNCI subsystem parameter specifies whether statements ignore a row that was inserted by another transaction if the row has not yet been detected as committed. A newly inserted row can be detected as committed only after the lock held on the row has been released.

Acceptable values: NO, YES
Default: NO
Update: option 29 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SKIPUNCI
SKIP UNCOMM INSERTS applies only to statements running with row-level locking and read stability or cursor stability isolation level.

**NO** Db2 waits for the inserted row to be committed or rolled back. It then processes the row if the insert commits, or it moves on to find another row if the insert is rolled back. If a transaction performs one or more inserts, and then spawns a second transaction, specify NO for SKIP UNCOMM INSERTS if the first transaction needs the second transaction to wait for the outcome of the inserts.

This is the default value.

**YES**

Until the lock held on a newly inserted row is released, Db2 behaves as though the newly inserted row has not yet arrived and the row is skipped. Specifying a value of YES offers greater concurrency than the default value of NO.

**Related tasks:**
- [Accessing currently committed data to avoid lock contention (Db2 Performance)]
- [Choosing a CURRENTDATA option (Db2 Performance)]
- [Choosing an ISOLATION option (Db2 Performance)]
- [Improving concurrency for applications that tolerate incomplete results (Db2 Performance)]

**Related reference:**
- [CONCURRENTACCESSRESOLUTION bind option (Db2 Commands)]

**IMMEDIATE WRITE field (IMMEDWRI subsystem parameter)**

The IMMEDWRI subsystem parameter determines when updates to group buffer pool-dependent buffers are to be written to the coupling facility.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>YES, NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>NO</td>
</tr>
<tr>
<td>Update:</td>
<td>option 29 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6GRP IMMEDWRI</td>
</tr>
</tbody>
</table>

**YES**

Db2 immediately writes the page to the coupling facility after the update occurs.

**NO**

Db2 does not immediately write the change buffer to the coupling facility. Instead, Db2 waits until phase 1 of commit.

If either the IMMEDWRITE bind option or the IMMEDWRI subsystem parameter is set to YES, the value of the immediate write option at run time is also YES.

**PLAN MANAGEMENT field (PLANMGMT subsystem parameter)**

The PLANMGMT subsystem parameter specifies which default plan management policy to use for storing information about access paths information for SQL statements when the PLANMGMT bind option is not explicitly specified for the rebind of a package.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>EXTENDED, OFF, BASIC,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>EXTENDED</td>
</tr>
<tr>
<td>Update:</td>
<td>option 29 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6SPRM PLANMGMT</td>
</tr>
</tbody>
</table>
EXTENDED
Db2 saves the active copy, and two older copies, which are known as the previous and original copies. Db2 replaces the previous copy with the former active copy and saves the newest copy as the active copy. Db2 saves the original copy at the first rebind that EXTENDED is specified and retains the original copy unchanged at subsequent rebinds. EXTENDED is the default value.

BASIC
Db2 saves the active copy and one older copy, which is known as the previous copy. Db2 replaces the previous copy path with the former active copy and saves the newest copy the active copy.

OFF
No copies are saved.

Related tasks:
 Saving and switching to previous access paths (Db2 Performance)

Related reference:
 PLAN MGMT bind option (Db2 Commands)

PLAN MANAGEMENT SCOPE field (PLANMGMTSCOPE subsystem parameter)
The PLANMGMTSCOPE subsystem parameter specifies the default plan management scope to use when the PLANMGMTSCOPE option is not explicitly specified for the bind or rebind of a package.

The value in this field is meaningful only when the value of the PLAN MANAGEMENT field is BASIC or EXTENDED.

Acceptable values:
STATIC

Default:
STATIC

Update:
option 29 on panel DSNTIPB

DSNZPxxx:
DSN6SPRM PLANMGMTSCOPE

ALL
Db2 stores access path information for static and dynamic SQL queries.

STATIC
Db2 stores access path information for only static SQL queries. This is the default value for the PLANMGMTSCOPE parameter.

Related tasks:
 Saving and switching to previous access paths (Db2 Performance)

PACKAGE RELEASE COMMIT field (PKGREL_COMMIT subsystem parameter)
The PKGREL_COMMIT subsystem parameter specifies whether, at COMMIT or ROLLBACK, a persistent Db2 thread will release a package that is active on that thread if certain Db2 operations are waiting for exclusive access to that package. This parameter is applicable to both local and distributed threads.

The value in this field is meaningful only for packages that are bound with the RELEASE(DEALLOCATE) bind option.

Acceptable values:
YES, NO
YES
For packages that are bound with the RELEASE(DEALLOCATE) option, the following operations are permitted at COMMIT or ROLLBACK while the package is active and allocated by Db2 for a persistent Db2 thread:

- BIND REPLACE PACKAGE and REBIND PACKAGE requests, including auto rebind online schema changes for tables and indexes that are statically referenced by the package
- Online REORG operations that materialize pending definition changes for objects that are statically referenced by the package

This is the default value for the PKGREL_COMMIT parameter.

NO
Db2 will not implicitly release an active package at COMMIT or ROLLBACK for a persistent Db2 thread.

Related reference:

RELEASE bind option (Db2 Commands)

RANDOMIZE XML DOCID field (XMLRANDOMIZEDOCID subsystem parameter)
Specifies whether Db2 is to sequentially or randomly generate the DOCID values for XML columns.

This parameter does not affect existing tables that have XML columns. Those tables continue to generate DOCIDs in the order that was specified by XMLRANDOMIZEDOCID when the table was created or in sequential order if the parameter was not specified.

Acceptable values: NO, YES
Default: NO
Update: option 29 on panel DSNTPB
DSNZPxxx: DSN6SYSP XMLRANDOMIZEDOCID

NO
Db2 generates the DOCID values for XML columns sequentially.

YES
Db2 generates the DOCID values for XML columns randomly. Setting this parameter to YES can be beneficial for performance when concurrent inserts are expected.

Once a table starts to randomly generate DOCID values for XML columns, it cannot be changed to generate the value sequentially. Changing this parameter back to NO does not have any effect on tables that have had columns added with a randomized DOCID.

DISABLE EDM RTS field (DISABLEEDMRTS subsystem parameter)
The DISABLE_EDMRTS subsystem parameter specifies whether to disable the collection of real-time statistics by the environmental description manager (EDM). By default, EDM collects real-time statistics to track when packages were last used.

Acceptable values: NO, YES
Default: NO
Update: option 29 on panel DSNTIPB
DSNZPxxx: DSN6SPRM DISABLE_EDMRTS

**NO** EDM continues to collect real-time statistics about the last use of packages.
This is the recommended value.

**YES**
EDM does not collect real-time statistics about the last use of packages. YES might be appropriate in the following situations:
- You have no need to collect information about the last use of packages.
- The SYSIBM.SYSPACKAGE table is persistently unavailable, which prevents the recording of EDM real-time statistics.
- Deferred externalization of these statistics to the SYSIBM.SYSPACKAGE table causes excessive Db2 shutdown time.
- Collection of these EDM statistics causes performance degradation.

In a data sharing environment, all members should use the same setting for this parameter.

**REAL TIME STATS field (STATSINT subsystem parameter)**
The STATSINT subsystem parameter specifies the interval, in minutes, at which Db2 writes in-memory page set statistics to the real-time statistics tables. The value of the STATSINT subsystem parameter also controls the interval for writing statistics recommendations to the SYSIBM.SYSSSTATSFEEDBACK catalog table. Db2 waits the specified number of minutes after the completion of the previous operation to begin the next operation.

Acceptable values: 1 - 1440
Default: 30
Update: option 24 on panel DSNTIPB
DSNZPxxx: DSN6SPRM STATSINT

Related concepts:
- When Db2 externalizes real-time statistics (Db2 Performance)

Related tasks:
- Controlling the collection of statistics feedback (Db2 Performance)

Related reference:
- SYSINDEXSPACESTATS catalog table (Db2 SQL)
- SYSTABLESPACESTATS catalog table (Db2 SQL)
- SYSSTATSFEEDBACK catalog table (Db2 SQL)

**STATISTICS FEEDBACK field (STATFDBK_SCOPE subsystem parameter)**
The STATFDBK_SCOPE subsystem parameter specifies the scope of the SQL statements that Db2 collects statistics recommendations for in the SYSIBM.SYSSSTATSFEEDBACK catalog table.

Acceptable values: ALL, DYNAMIC, NONE, or STATIC
Default: ALL
Update: option 24 on panel DSNTIPO
DSNZPxxx: DSN6SPRM STATFDBK_SCOPE
NONE
No statistics recommendations are collected in the catalog table.

DYNAMIC
Statistics recommendations are collected in the catalog table for dynamic SQL statements only.

STATIC
Statistics recommendations are collected in the catalog table for static SQL statements only.

ALL
Statistics recommendations are collected in the catalog table for all SQL Statements.

Related tasks:
- Controlling the collection of statistics feedback (Db2 Performance)

Related reference:
- SYSTFAFEEDBACK catalog table (Db2 SQL)

STATISTICS PROFILE FEEDBACK field (STATFDBK_PROFILE subsystem parameter)
The STATFDBK_PROFILE subsystem parameter specifies whether statistics recommendations identified during query optimization directly result in modifications to statistics profiles.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: YES or NO
Default: Yes
Data sharing scope: It is recommended that all members use the same setting.
Update: option 24 on panel DSNTIPO
DSN6SPRM.STATFDBK_PROFILE

YES
Db2 modifies or creates statistics profiles based on statistics recommendations during query optimization.

Whenever Db2 writes new statistics recommendations to the SYSTFDBK catalog table, it also updates existing statistics profiles based on the recommendations:
- If no profile exists, Db2 creates new profiles for some recommendations. The new profiles collect existing statistics (similar to the result of RUNSTATS SET PROFILE FROM EXISTING STATS) plus statistics for new and existing uncollected statistics recommendations.
- Existing profiles are modified depending on the type of statistics recommendation:
  - TYPE='T'
    No profiles are changed, and no profiles are created for this recommendation type.
  - TYPE='I'
    If the existing statistics profile does not include the index, the index is added with default collection. Any existing list of indexes without FREQVAL or HISTOGRAM specified is changed to INDEX(ALL).
No profiles are created for this recommendation type.

**TYPE=C' with NUMCOLS = 1**

If the existing profile does not include the column, the column is added.

No profiles are created for this recommendation type.

**TYPE=C' with NUMCOLS > 1**

If the existing statistics profile does not include the column group, the column group is added with COLGROUP syntax. KEYCARD recommendations add the corresponding index.

If no statistics profile exists for the table, a new profile that includes the column group and INDEX(ALL) is created.

**TYPE=F' or 'H'**

If the existing statistics profile does not include the frequency or histogram statistic, it is added. Db2 automatically determines the number columns to collect. Histogram statistics are added without a NUMQUANTILES specification. The frequency or histogram statistics are added in COLGROUP syntax. However, if the recommendation is for an index on expression, INDEX syntax is used instead.

In no statistics profile exists for the table, a new profile that collects the frequency or histogram statistics and INDEX(ALL) is created.

- If Db2 detects certain statistics that provide no value for query optimization, such as certain multi-column frequency statistics, it modifies the profile to prevent the future collection of the useless statistics. Db2 also removes the statistics from the relevant catalog tables and issues a message to indicate the removal.
- If CREATE INDEX statement is issued, Db2 adds the index to the existing statistics profile for the related table.

**NO** Db2 does not modify statistics profiles based on statistics recommendations during query optimization. However, Db2 might still write the recommendations to the SYSIBM.SYSSTATFEEDBACK catalog table, depending on the value of subsystem parameter STATFDBK_SCOPE.

Related concepts:

- [Statistics profiles (Db2 Performance)](Related_concepts)

Related tasks:

- [Applying statistics recommendations to statistics profiles automatically (Db2 Performance)](Related_tasks)
- [Controlling the collection of statistics feedback (Db2 Performance)](Related_tasks)

**DSNTIP81: Performance and optimization panel 2**

The DSNTIP81 panel is a continuation of the DSNTIP8 panel. It is used to set application programming default values pertaining to performance and optimization panel.
CURRENT DEGREE field (CDSSRDEF subsystem parameter)

The CDSSRDEF subsystem parameter determines the default value that is to be used for the CURRENT DEGREE special register. The default value is used when a degree is not explicitly set in the SQL statement SET CURRENT DEGREE.

Acceptable values: 1, ANY
Default: 1
Update: option 30 on panel DSNTIPB
DSNZPxxx: DSN6SPRM CDSSRDEF

1 Specifies that when a query is dynamically prepared, the execution of that query will not use parallelism.
If this value is specified, Db2 does not use any optimization hints for parallelism.

ANY Specifies that when a query is dynamically prepared, the execution of that query can involve parallelism.

Recommendation: In almost all situations, accept the default value of 1. You should use parallelism selectively where it provides value, rather than globally. Although parallelism can provide a substantial reduction in elapsed time for some queries with only a modest overhead in processing time, parallelism does not always provide the intended benefit. For some queries and in many other situations, query parallelism does not provide an improvement, or it uses too many resources. If you are using nearly all of your CPU, I/O, or storage resources, parallelism is more likely to cause degradation of performance. Use parallelism only where it is most likely to provide benefits.

MAX DEGREE field (PARAMDEG subsystem parameter)

The PARAMDEG subsystem parameter specifies the maximum degree of parallelism for a parallel group. When you specify a non-zero value for this parameter, you limit the degree of parallelism so that Db2 cannot create too many parallel tasks that use virtual storage.
Acceptable values: 0 - 254
Default: 0
Update: option 30 on panel DSNTIPB

0  Specifies no limit to the maximum degree of parallelism that Db2 chooses based on the cost estimate for the query and the system configuration, in particular the number of processors online. Db2 counts both general purpose and zIIP processors equally, and applies further adjustment to determine the degree to use.

1 - 254  Specifies the maximum degree of parallelism that Db2 uses.

When optimization hints for parallelism are used, the value of the PARAMDEG subsystem parameter does not limit the degree of parallelism at bind time. However, the value of the PARAMDEG subsystem parameter is enforced at execution time. So, if the value of the PARAMDEG subsystem parameter is lower than the degree of parallelism that is specified at bind time, the degree of parallelism is reduced at execution time.

Tip: For systems with more than two zIIP processors configured, use the number of zIIP processors as the starting value, and then adjust as needed for your response time requirements.

Related tasks:
- [Enabling parallel processing (Db2 Performance)](#)
- [Disabling query parallelism (Db2 Performance)](#)
- [Specifying optimization parameters at the statement level (Db2 Performance)](#)

**MAX DEGREE FOR DPSI (PARAMDEG_DPSI subsystem parameter)**

The PARAMDEG_DPSI subsystem parameter specifies the maximum degree of parallelism that you can specify for a parallel group in which a data partitioned secondary index (DPSI) is used to drive parallelism.

A DPSI is a nonpartitioning index that is physically partitioned according to the partitioning scheme of the table. When you specify a value of greater than 0 for this parameter, you limit the degree of parallelism for DPSIs so that Db2 does not create too many parallel tasks that use virtual storage.

Acceptable values: 0-254, DISABLE
Default: 0
Update: option 30 on panel DSNTIPB

0  Specifies that Db2 uses the value that is specified for the PARAMDEG subsystem parameter, instead of PARAMDEG_DPSI, to control the degree of parallelism when DPSI is used to drive parallelism. This is the default value for the field.

1  Specifies that Db2 creates multiple child tasks but works on one task at a time when DPSI is used to drive parallelism.
Specifies that Db2 creates multiple child tasks and works concurrently on the tasks that are specified. The number of specified tasks may be larger or smaller than the number of tasks as specified in PARAMDEG. When PARAMDEG is set to 1, the rest of the query does not have any parallelism.

DISABLE
Specifies that Db2 does not use DPSI to drive parallelism. Parallelism might still occur for the query if PARAMDEG is greater than 1.

PARALLELISM EFFICIENCY field (PARA_EFF subsystem parameter)
The PARA_EFF subsystem parameter controls the efficiency that Db2 assumes for parallelism when Db2 chooses an access path. The integer value that is used for this parameter represents a percentage efficiency.

Acceptable values: 0 – 100
Default: 50
Update: option 30 on panel DSNTIPB
DSNZPxxx: DSN6SPRM PARA_EFF

Related tasks:
Tuning parallel processing (Db2 Performance)

STAR JOIN QUERIES field (STARJOIN subsystem parameter)
The STARJOIN subsystem parameter specifies whether star join processing is to be enabled.

Acceptable values: DISABLED, ENABLED, 1 - 32768
Default: DISABLED
Update: option 30 on panel DSNTIPB
DSNZPxxx: DSN6SPRM STARJOIN

DISABLED
Star schema processing is disabled.

ENABLED
Star schema processing is enabled if the cardinality of the fact table is at least 25 times the cardinality of the largest dimension that is a base table that is joined to the fact table.

1 Star schema processing is enabled. The single table with the largest cardinality is the fact table. However, if more than one table has this cardinality, star join is not enabled.

2 - 32768 Star schema processing is enabled if the cardinality of the fact table is at least the specified number times the cardinality of the largest dimension that is a base table that is joined to the fact table.

Related concepts:
Star schema access (Db2 Performance)

Related reference:
SJTABLES in macro DSN6SPRM
MAX DATA CACHING field (MXDTCACH subsystem parameter)
The MXDTCACH subsystem parameter specifies the maximum amount of memory, in MB, that is to be allocated for data caching per thread.

Acceptable values: 0 - 512
Default: 20
Update: option 30 on panel DSNTIPB
DSNZPxxx: DSN6SPRM MXDTCACH

If you specify 0, Db2 does not use data caching during query execution. If you specify a value between 1 and 512, Db2 allocates memory from above the 2 GB bar pool up to the specified size for data caching.

Related concepts:
- Nested loop join (METHOD=1) (Db2 Performance)

CURRENT REFRESH AGE field (REFSHAGE subsystem parameter)
The REFSHAGE subsystem parameter specifies the default value for the CURRENT REFRESH AGE special register. The default value is used when no value is explicitly set by using the SET CURRENT REFRESH AGE statement.

Acceptable values: 0, ANY
Default: 0
Update: option 30 on panel DSNTIPB
DSNZPxxx: DSN6SPRM REFSHAGE

Accepting the default value of 0 disables query rewrite using deferred materialized query tables.

CURRENT MAINT TYPES field (MAINTYPE subsystem parameter)
The MAINTYPE subsystem parameter specifies the default value for the CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION special register. This value is to be used when no value is explicitly set by using the SQL statement SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION.

Acceptable values: NONE, SYSTEM, USER, ALL
Default: SYSTEM
Update: option 30 on panel DSNTIPB
DSNZPxxx: DSN6SPRM MAINTYPE

NONE
- Does not allow query rewrite.

SYSTEM
- Allows query rewrite by using system-maintained materialized query tables (SYSTEM) when CURRENT REFRESH AGE is set to ANY.

USER
- Allows query rewrite by using user-maintained materialized query tables when CURRENT REFRESH AGE is set to ANY.
ALL

Allows query rewrite by using both system-maintained and user-maintained materialized query tables.

**DSNTP82: Query accelerator preferences panel**

The DSNTP82 panel is used to set application programming default values pertaining to query accelerators.

---

**ACCELERATION STARTUP field (ACCEL subsystem parameter)**

Specifies whether accelerator servers can be used with a Db2 subsystem, and how the accelerator servers are to be enabled and started. An accelerator server cannot be started unless it is enabled.

**Acceptable values:** NO, AUTO, COMMAND

**Default:** NO

**Update:** option 31 on panel DSNTPB

**DSNZPxxx:** DSN6SPRM.ACCEL

**NO** Specifies that accelerator servers cannot be used with the Db2 subsystem.

**AUTO** Specifies that accelerator servers are automatically enabled and started when the Db2 subsystem is started.

**COMMAND** Specifies that accelerator servers are automatically enabled when the Db2 subsystem is started. The accelerator servers can be started with the Db2 START ACCEL command.

**GET ACCEL ARCHIVE field (GET_ACCEL_ARCHIVE subsystem parameter)**

The GET_ACCEL_ARCHIVE subsystem parameter determines the default value that is to be used for the CURRENT GET_ACCEL_ARCHIVE special register.
The GET_ACCEL_ARCHIVE subsystem parameter is used when no value is explicitly set for the CURRENT GET_ACCEL_ARCHIVE special register by the SQL statement SET CURRENT GET_ACCEL_ARCHIVE.

Acceptable values: NO, YES
Default: NO
Update: option 31 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.GET_ACCEL_ARCHIVE

NO Specifies that if a table is archived in an accelerator server, and a query references that table, the query does not use the data that is archived.
YES Specifies that if a table is archived in an accelerator server, and a query references that table, the query uses the data that is archived.

ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter)
The QUERY_ACCEL_OPTIONS subsystem parameter controls whether certain types of queries are allowed to execute on an accelerator server.

Acceptable values: NONE, YES
Default: NONE
DSNZPxxx: DSN6SPRM.QUERY_ACCEL_OPTIONS

Choose NONE if you do not want to choose whether Db2 allows certain types of queries to run on an accelerator server. If you choose NONE, Db2 does not allow any of those types of queries to run on an accelerator server. If you choose YES, you proceed to panel DSNTIP8A after completing panel DSNTIP82.

This panel is displayed:

```
+------------------------ ACCELERATION OPTIONS ------------------------+
DSNTIP8A Acceleration Options Panel
===>
Allow queries to run on an accelerator server that contain:
1 MBCS EBCDIC       ===> NO  NO or YES
2 INSERT FROM SELECT ===> NO  NO or YES
3 BYTE-BASED STRING FUNCTIONS ===> NO  NO or YES
4 DD/MM/YYYY LOCAL DATE FORMAT ===> NO  NO or YES
5 SYSTEM TIME TEMPORAL QUERIES ===> NO  NO or YES
6 TS COLS WITH PRECISION OF 12 ===> NO  NO or YES
7 YYYYMMDD LOCAL DATE FORMAT ===> NO  NO or YES
8 FAVOR NEW ACCELERATOR_TYPE ===> NO  NO or YES

PRESS: ENTER to continue  UP/DOWN to scroll  RETURN to exit
+----------------------------------------------------------------------+
```

Figure 36. DSNTIP8A

MBCS EBCDIC

Acceptable values: NO, YES
Default: NO
DSNZPxxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 1)
**NO** Specifies that queries that include data that is encoded in a multi-byte-character-set EBCDIC encoding scheme are blocked from executing on an accelerator server.

**YES**

Specifies that queries that include data that is encoded in a multi-byte-character-set EBCDIC encoding scheme are allowed to execute on an accelerator server.

If the accelerator server encodes data in another encoding scheme, such as Unicode UTF-8, the collating sequence for the data on the accelerator server is different from the collating sequence for the data in Db2. This difference affects data ordering and the results from range predicates. Therefore, if table columns include any mixture of uppercase characters, lowercase characters, numeric characters, or national language characters, and the SQL statements that query those tables include range predicates or ordering on those columns, the results of executing a query in Db2 might be different from the results of executing the same query on an accelerator server.

**INSERT FROM SELECT**

Acceptable values: NO, YES

Default: NO

DSNZPxxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 2)

**NO** Specifies that for queries that include an INSERT FROM SELECT statement, the SELECT operation does not run on an accelerator server.

**YES**

Specifies that for queries that include an INSERT FROM SELECT statement, the SELECT operation can run on an accelerator server.

If the SELECT operation runs on the accelerator server, the data that is operated on by the SELECT operation might not be current in the accelerator server.

**BYTE-BASED STRING FUNCTIONS**

Acceptable values: NO, YES

Default: NO

DSNZPxxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 3)

**NO** Specifies that queries that contain built-in functions for which Db2 processes each byte of the input string, rather than each character of the input string, do not run on an accelerator server.

**YES**

Specifies that queries that contain built-in functions for which Db2 processes each byte of the input string, rather than each character of the input string, can run on an accelerator server. The queries can run on an accelerator server even when the input string is encoded in a multi-byte-character encoding scheme. Unicode is an example of a multi-byte-character encoding scheme.

For an accelerator server that supports only functions that operate on the characters in an input string, if the input data on which the function operates contains only single-byte characters, the function returns the same result when it runs on the accelerator server and when it runs on Db2. However, if the input data contains multi-byte characters, the function returns different results when it runs on the accelerator server and when it runs on Db2.
**DD/MM/YYYY LOCAL DATE FORMAT**

Acceptable values: NO, YES  
Default: NO  
DSNZP.xxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 4)

**NO** Specifies that queries that reference an expression with a DATE data type that uses a LOCAL format are blocked from running on an accelerator server.

**YES**  
Specifies that queries that reference an expression with a DATE data type that uses a LOCAL format are not blocked from running on an accelerator server.  
The accelerator server uses the format 'dd/mm/yyyy' to interpret the input and output date value.

Option 4 is not permitted when option 7 has been selected.

Specify option 4 only when one of the following conditions exists:
- The DATE FORMAT field of the DSNTIP4 panel specifies LOCAL.  
- Application programs that process SQL on Db2 are precompiled with the DATE(LOCAL) option.

In both of these cases, the LOCAL date exit routine must define the specific 'dd/mm/yyyy' date format. If the LOCAL format is not defined as 'dd/mm/yyyy', the query might return unpredictable results.

**SYSTEM_TIME TEMPORAL QUERIES**

Acceptable values: NO, YES  
Default: NO  
DSNZP.xxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 5)

**NO** Specifies that SYSTEM_TIME temporal queries will not be offloaded to an accelerator.

**YES**  
Specifies that SYSTEM_TIME temporal queries will be offloaded to an accelerator.

**TS COLS WITH PRECISION OF 12**

Acceptable values: NO, YES  
Default: NO  
DSNZP.xxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 6)

**NO** Specifies that queries will not be offloaded to an accelerator if the queries reference timestamp columns with precision of 12.

**YES**  
Specifies that queries will be offloaded to an accelerator if the queries reference timestamp columns with precision of 12.

**YYYYMMDD LOCAL DATE FORMAT**

Acceptable values: NO, YES  
Default: NO  
DSNZP.xxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 7)
NO  Specifies that queries that reference an expression with a DATE data type that uses a LOCAL format are blocked from running on an accelerator server.

YES  Specifies that queries that reference an expression with a DATE data type that uses a LOCAL format are not blocked from running on an accelerator server. The accelerator server uses the format YYYYMMDD, padded with two blanks at the end, to interpret the input and output data value.

Option 7 is not permitted when option 4 has been selected.

Specify option 7 only when one of the following conditions exists:

- The DATE FORMAT field of the DSNTIP4 panel specifies LOCAL.
- Application programs that process SQL on Db2 are precompiled with the DATE(LOCAL) option.

In both cases, the LOCAL date exit routine must define the specific 'yyyyymmdd' date format. If the LOCAL format is not defined as 'yyyyymmdd', the query might return unpredictable results.

FAVOR NEW ACCELERATOR_TYPE
Acceptable values: NO, YES
Default: NO
DSNZPxxx: DSN6SPRM.QUERY_ACCEL_OPTIONS (option 8)

NO  Specifies that in a Db2 environment that includes both a V5 accelerator server and a V7 accelerator server, statements that are eligible to be accelerated on either of the accelerator servers are sent to the V5 accelerator server.

YES  Specifies that in a Db2 environment that includes both a V5 accelerator server and a V7 accelerator server, statements that are eligible to be accelerated on either of the accelerator servers are sent to the V7 accelerator server.

CURRENT QUERY ACCEL field (QUERY_ACCELERATION subsystem parameter)
The QUERY_ACCELERATION subsystem parameter determines the default value that is to be used for the CURRENT QUERY ACCELERATION special register.

The QUERY_ACCELERATION subsystem parameter is used when no value is explicitly set for the CURRENT QUERY ACCELERATION special register by the SQL statement SET CURRENT QUERY ACCELERATION.

Acceptable values: 1, 2, 3, 4, 5
Default: 1
Update: option 31 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.QUERY_ACCELERATION

1 (NONE)
   Specifies that no query acceleration is done.

2 (ENABLE)
   Specifies that queries are accelerated only if Db2 determines that it is advantageous to do so. If there is an accelerator failure while a query is running, or the accelerator returns an error, Db2 returns a negative SQLCODE to the application.
3 (ENABLE_WITH_FAILBACK)
Specifies that queries are accelerated only if Db2 determines that it is advantageous to do so. If the accelerator returns an error during the PREPARE or first OPEN for the query, Db2 executes the query without the accelerator. If the accelerator returns an error during a FETCH or a subsequent OPEN, Db2 returns the error to the user, and does not execute the query.

Restriction: [FL 504] If the query contains a passthrough-only expression, Db2 returns an error and does not accelerate the query, even if a matching user-defined function exists. For more information about passthrough-only expressions, see Accelerating queries with passthrough-only expressions.

4 (ELIGIBLE)
Specifies that queries are accelerated if they are eligible for acceleration. Db2 does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are executed by Db2. If there is an accelerator failure while a query is running, or the accelerator returns an error, Db2 returns a negative SQLCODE to the application.

5 (ALL)
Specifies that queries are accelerated if they are eligible for acceleration. Db2 does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are not executed by Db2, and a SQL error is returned. If there is an accelerator failure while a query is running, or the accelerator returns an error, Db2 returns a negative SQLCODE to the application.

Related concepts:
- How Db2 determines whether to accelerate eligible queries (Db2 Performance)

WAIT FOR DATA field (QUERY_ACCEL_WAITFORDATA subsystem parameter)
The QUERY_ACCEL_WAITFORDATA subsystem parameter determines the default value that is to be used for the CURRENT QUERY ACCELERATION WAITFORDATA special register.

The QUERY_ACCEL_WAITFORDATA subsystem parameter is used when no value is explicitly set for the CURRENT QUERY ACCELERATION WAITFORDATA special register by the SQL statement SET CURRENT QUERY ACCELERATION WAITFORDATA.

Acceptable values: 0.0 - 3600.0 or 0 - 3600
Default: 0.0
Update: option 31 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.QUERY_ACCELERATION_WAITFORDATA

Valid values are a DECIMAL(5,1) numeric constant in the range of 0.0 - 3600.0 seconds. For example, a value of 20.0 represents 20.0 seconds (or 20000 milliseconds), and a value of 30.5 represents 30.5 seconds (or 30500 milliseconds). The maximum value of 3600.0 means that the query is delayed for 60 minutes.

The default value is 0.0, which means that the query is not delayed and is run immediately on the accelerator. Other WAITFORDATA behavior is not applied to the query.
The wait time value can also be specified as an INTEGER numeric constant value ranging from 0 - 3600 seconds, which Db2 will convert to a DECIMAL (5,1) value.

When a non-zero value is specified for QUERY_ACCEL_WAITFORDATA, Db2 and the accelerator will apply other WAITFORDATA delay behavior, restrictions, and requirements to all dynamic queries to be accelerated. These behaviors, restrictions, and requirements can validly stop or even fail the acceleration of some queries that formerly were accelerated successfully. If you specify a non-zero WAITFORDATA delay on the new Db2 subsystem parameter, all dynamic queries on the Db2 subsystem for which acceleration is requested will be affected. For more information about these behaviors, restrictions, and requirements, see SET CURRENT QUERY ACCELERATION WAITFORDATA (Db2 SQL).

Attention: It is recommended that you do not use the QUERY_ACCEL_WAITFORDATA subsystem parameter to provide an initial non-zero default value for CURRENT QUERY ACCELERATION WAITFORDATA special register unless you fully understand and can accept the possible impact of WAITFORDATA behavior, requirements, and restrictions on all dynamic queries to be accelerated on the Db2 subsystem.

Related reference:
- SET CURRENT QUERY ACCELERATION WAITFORDATA (Db2 SQL)
- CURRENT QUERY ACCELERATION WAITFORDATA (Db2 SQL)
- CURRENT QUERY ACCELERATION (Db2 SQL)

DSNTIPI: IRLM panel 1

The entries on the DSNTIPI panel affect the installation of the internal resource lock manager (IRLM).

You must use one IRLM for each Db2 subsystem.

Figure 37. IRLM panel 1: DSNTIPI

Related concepts:
- IRLM address space (IRLPROC) storage requirements
INSTALL IRLM field
The INSTALL IRLM field determines whether IRLM subsystem entries are to be provided in job DSNTIJMV and whether an IRLM procedure is to be built.

Acceptable values: YES, NO
Default: YES
Update: Edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

DSNZPxxx: none

YES
The required entries are provided, and the IRLM procedure is built.

NO
No IRLM procedure is produced. On installation panel DSNTIPJ, all values are ignored except the LOCKS PER TABLE(SPACE) field and the LOCKS PER USER field.

If you do not have a new IRLM procedure created, ensure that your old IRLM procedure is updated with any new keywords that were added.

SUBSYSTEM NAME field (IRLMSID subsystem parameter)
The IRLMSID subsystem parameter specifies the name by which z/OS is to know the IRLM subsystem.

Acceptable values: 1 - 4 characters. First character must be A - Z, #, $, or @. Other characters must be A - Z, 1 - 9, #, $, or @.
Default: IRLM
Update: option 32 on panel DSNTIPB. Also edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

DSNZPxxx: DSN6SPRM IRLMSID

The name is used for communication between Db2 and the IRLM. This name is included in the z/OS subsystem table IEFSSNxx, where xx is the value of the SUBSYSTEM MEMBER field on installation panel DSNTIPM.

If you installed the IRLM for IMS, The IRLM name for Db2 must be different. Two IRLMs that reside in the same z/OS system must have unique z/OS subsystem names. If you already have IRLM installed, use the z/OS subsystem name for that IRLM. Otherwise, accept the default value, IRLM.

The associated IRLM PROC parameter is IRLMNM.

Related reference:
COMMAND PREFIX field

RESOURCE TIMEOUT field (IRLMRWT subsystem parameter)
The IRLMRWT subsystem parameter controls the number of seconds that are to elapse before a resource timeout is detected.

Acceptable values: 1 - 3600
Timeout means that a lock request has waited for a resource (or for claims on a resource for a particular claim class to be released) longer than the number of seconds specified on this option. The value that is specified for this option must be a multiple of the DEADLOCK TIME on installation panel DSNTIPJ because IRLM uses its deadlock timer to initiate timeout detection and deadlock detection. This value is rarely the actual time. For data sharing, the actual timeout period is longer than the timeout value.

Related concepts:
- Global timeout processing (Db2 Data Sharing Planning and Administration)

Related tasks:
- Specifying the amount of inactive time before a timeout (Db2 Performance)

Related reference:
- DEADLOCK TIME field
- DATA DEF TIMEOUT field (DDLTOX subsystem parameter)

AUTO START field (IRLMAUT subsystem parameter)
The IRLMAUT subsystem parameter defines whether Db2 is to automatically start and stop the IRLM.

Acceptable values: YES, NO
Default: YES
Update: option 32 on panel DSNTIPB
DSNZPxxx: DSN6SPRM IRLMAUT

**YES**
When Db2 starts, it tries to start the IRLM if the IRLM is not already started. When Db2 stops, IRLM automatically stops if the IRLM was started by Db2. However, IRLM will not automatically terminate if it is defined with DISCONNECT IRLM = NO.

**NO** Db2 terminates if the IRLM is not started when Db2 comes up.

**Recommendation:** Use YES if you use the IRLM for only a single Db2 subsystem.

When IRLM initializes, it is registered with the z/OS automatic restart manager (ARM). IRLM deregisters from the ARM when the IRLM is shut down normally. When IRLM terminates, it sends Db2 the registration information so that Db2 can determine whether IRLM was terminated normally. Db2 then deregisters from the ARM to prevent unwanted restarts. Otherwise, IRLM might be restarted with Db2, if AUTO START is YES.

**Related concepts:**
- IRLM names

**Related tasks:**
- Controlling the IRLM (Db2 Administration Guide)

PROC NAME field (IRLMPRC subsystem parameter)
The IRLMPRC subsystem parameter specifies the name of the IRLM procedure that z/OS is to invoke if the AUTO START field is set to YES.
Acceptable values: 1 - 8 characters
Default: IRLMPROC
Update: option 32 on panel DSNTIPB. Also edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

DSNZPxxx: DSN6SPRM IRLMPROC

The name that is specified in the PROC NAME field cannot be the same as the subsystem name that is specified in the SUBSYSTEM NAME field.

The procedure is created during installation or migration by job DSNTIJMV and is placed in SYS1.PROCLIB. You can review the procedure by examining DSNTIJMV.

**TIME TO AUTOSTART field (IRLMSWT subsystem parameter)**
The IRLMSWT subsystem parameter specifies the IRLM wait time in seconds. This is the amount of time that Db2 is to wait for the IRLM to start during autostart. If the time expires, Db2 abends.

Acceptable values: 1 - 3600
Default: 120
Update: option 32 on panel DSNTIPB

DSNZPxxx: DSN6SPRM IRLMSWT

**U LOCK FOR RR/RS field (RRULOCK subsystem parameter)**
When operations use repeatable read (RR) or read stability (RS) isolation levels to access a table, a specified locking method is used. The RRULOCK subsystem parameter specifies whether Db2 is to use U (UPDATE) locks or S (SHARE) locks.

Acceptable values: YES, NO
Default: YES
Update: option 32 on panel DSNTIPB

DSNZPxxx: DSN6SPRM RRULOCK

**YES**
The lock mode for operations with RR or RS isolation levels is U. If your applications make frequent updates with repeatable-read isolation, the U-lock might provide greater concurrency than the S-lock. However, applications that require high concurrency are almost always more efficient if they use cursor stability (CS) isolation.

**NO**
The lock mode for operations with RR or RS isolation levels is S. If the cursor in your applications includes the FOR UPDATE OF clause, but updates are infrequent, S-locks generally provide better performance.

Related concepts:

- [How Db2 chooses lock types (Db2 Performance)](index)
- [The ISOLATION (CS) option (Db2 Performance)](index)
- [The ISOLATION (RR) option (Db2 Performance)](index)
- [The ISOLATION (RS) option (Db2 Performance)](index)

Related tasks:

- [Improving concurrency for update and delete operations (Db2 Performance)](index)

Related reference:

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST 235
X LOCK FOR SEARCHED U/D field (XLKUPDLT subsystem parameter)

The XLKUPDLT subsystem parameter specifies the locking method that is to be used when Db2 executes a searched update or delete.

Acceptable values: YES, NO, TARGET

Default: NO

Update: option 32 on panel DSNTIPB

DSNZPxxx: DSN6SPRM XLKUPDLT

YES

Db2 uses an X-lock on qualifying rows or pages. For ISOLATION(CS), the lock is released if the rows or pages are not updated or deleted. For ISOLATION(RS) or ISOLATION(RR), an X-lock is retained until the next commit point. A value of YES is beneficial in a data sharing environment when most or all searched updates and deletes use an index. If YES is specified and searched updates or deletes result in a table space scan, the likelihood of timeouts and deadlocks greatly increases.

NO

Db2 uses an S- or U-lock when scanning for qualifying rows. For any qualifying rows or pages, the lock is upgraded to an X-lock before performing the update or delete. For non-qualifying rows or pages the lock is released if ISOLATION(CS) is used. For ISOLATION(RS) or ISOLATION(RR), an S-lock is retained on the rows or pages until the next commit point. Use this option to achieve higher rates of concurrency.

TARGET

Db2 combines YES and NO behavior. Db2 uses an X-lock on qualifying rows or pages of the specific table that is targeted by the update or delete statement. Db2 uses an S- or U-lock when scanning for rows or pages of other tables that are referenced by the query (for example, tables that are referenced only in the WHERE clause of the query). For non-qualifying rows or pages the lock is released if ISOLATION(CS) is used. For ISOLATION(RS) or ISOLATION(RR), an S-lock is retained on the rows or pages until the next commit point.

Related tasks:

Disabling update locks for searched UPDATE and DELETE (Db2 Data Sharing Planning and Administration)

Improving concurrency for update and delete operations (Db2 Performance)

START IRLM CTRACE field

The value of the START IRLM CTRACE field specifies whether the IRLM component traces are to be activated when IRLM is started. The Db2-provided IRLM procedure in DSNTIJMV is tailored according to this value.

Acceptable values: NO, YES

Default: NO

Update: Edit the associated parameter in job DSNTIJUZ, the IRLM start procedure, and the CLIST input member.

DSNZPxxx: none

NO IRLM starts only the low-activity subtraces EXP, INT, and XIT.
IRLM starts with all its subtraces active. Starting all IRLM subtraces has a slight impact on performance demand for ECSA storage, but it improves serviceability.

**IMS BMP TIMEOUT field (BMPTOUT subsystem parameter)**
The BMPTOUT subsystem parameter is used to determine the amount of time that an IMS BMP connection is to wait for a lock to be released. The value of this field is multiplied by the value specified in the RESOURCE TIMEOUT field to determine the wait time.

For example, if you use the default value, an IMS BMP connection can wait for a resource for four times as long as the resource timeout value. This option gives you flexibility in tuning your system to avoid timeouts.

Acceptable values: 1 - 254
Default: 4
Update: option 32 on panel DSNTIPB
DSNZPxxx: DSN6SPRM BMPTOUT

**Related tasks:**
- [Setting installation options for wait times (Db2 Performance)](#)

**DL/I BATCH TIMEOUT field (DLITOUT subsystem parameter)**
The DLITOUT subsystem parameter controls the number of resource timeout values that a DL/I batch connection is to wait for a lock to be released. The resource timeout value is specified in the RESOURCE TIMEOUT field.

Acceptable values: 1 - 254
Default: 6
Update: option 32 on panel DSNTIPB
DSNZPxxx: DSN6SPRM DLITOUT

For example, if you use the default value of 6, a DL/I batch application can wait 6 times the resource timeout value for a resource. This option gives you flexibility in tuning your system to avoid timeouts.

**Related tasks:**
- [Setting installation options for wait times (Db2 Performance)](#)

**Related reference:**
- [RESOURCE TIMEOUT field (IRLMRWT subsystem parameter)](#)

**RETAINED LOCK TIMEOUT field (RETLWAIT subsystem parameter)**
When another Db2 subsystem in a data sharing group has failed, it can hold an incompatible lock on a resource. Locks that are held by failed Db2 members are called retained locks. The RETLWAIT subsystem parameter is used to calculate how long a transaction is to wait for a retained lock.

Acceptable values: 0 - 254
Default: 0
Update: option 32 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RETLWAIT
This value is of importance only in a data sharing environment.

The value that you use is a multiplier that is applied to the normal timeout value of the connection. For example, imagine that the retained lock multiplier is 2. The timeout period for a call attachment connection that is waiting for a retained lock is 1 * 2 (1 for the normal CAF timeout period, 2 for the additional time that is specified for retained locks).

If you use the default value, 0, applications do not wait for incompatible retained locks. Instead, the lock request is immediately rejected, and the application receives a resource unavailable SQLCODE.

Related concepts:
- Retained locks (Db2 Data Sharing Planning and Administration)

DSNTIPJ: IRLM panel 2

The entries on the DSNTIPJ panel affect several characteristics of IRLM time-sharing fields and other locking options.

The default values are adequate for most sites under ordinary conditions. Db2 and IRLM group names must start with a letter.

![Figure 38. IRLM panel 2: DSNTIPJ](image)

Related concepts:
- IRLM address space (IRLMPROC) storage requirements
- Performance monitoring and tuning for data sharing environments (Db2 Data Sharing Planning and Administration)

Related reference:
- DSNTIPI: IRLM panel 1

PAGE PROTECT field

The value of the PAGE PROTECT field specifies whether IRLM is to load its common storage modules into page-protected storage.
Acceptable values: NO, YES
Default: YES
Update: edit IRLM start procedure
DSNZPxxx: none

NO Indicates that common storage modules are to be loaded into CSA or ECSA without page protecting that memory.

YES Indicates that modules located in common storage are to be loaded into page-protected storage to prevent programs from overlaying the instructions. YES is recommended because it requires no additional overhead after the modules are loaded, and the protection can prevent code-overlay failures.

This is the default value.

The associated IRLM PROC parameter is PGPROT.

MAX STORAGE FOR LOCKS field
The MAX STORAGE FOR LOCKS field specifies, in MB, the maximum amount of private storage above the 2 GB bar for the IRLM lock control block structure.

Acceptable values: 2048 - 102400
Default: 2160
Update: edit IRLM start procedure
DSNZPxxx: none

This value becomes the setting of the MLMT parameter for the IRLM address space procedure. The IRLM address space procedure sets the z/OS MEMLIMIT value for the address space.

Ensure that you set this value high enough so that IRLM does not reach the limit. The value that you choose should provide space for possible retained locks. IRLM only gets storage as it needs it, so choose a large value. You can also change the value dynamically by using the z/OS command MODIFY irlmproc,SET,MLT.

The associated IRLM PROC parameter is MLMT.

Related concepts:
Common service area storage requirements

Related reference:
IRLM storage use

LOCKS PER TABLE(SPACE) field (NUMLKTS subsystem parameter)
The NUMLKTS subsystem parameter specifies the default maximum number of page, row, or LOB locks that an application can hold simultaneously in a table or table space. If a single application exceeds the maximum number of locks in a single table or table space, lock escalation occurs.

Acceptable values: 0 - 104857600
Default: 2000
Update: option 33 on panel DSNTIPB
DSNZPxxx: DSN6SPRM NUMLKTS
You can enter the number of locks as an integer or you can enter a value with a suffix of K or M. If you enter a value with a suffix of K, the number of locks is that value multiplied by 1024. If you enter a value with a suffix of M, the number of locks is that value multiplied by 1024 × 1024, or 1,048,576.

The value that you specify for this field must be less than the value that is specified for the LOCKS PER USER field (except when LOCKS PER USER is set to 0).

The value of the LOCKS PER TABLE(SPACE) field becomes the default value (SYSTEM) for the LOCKMAX clause of the SQL statements CREATE TABLESPACE and ALTER TABLESPACE. A value of 0 indicates that there is no limit to the number of data and row locks that a program can acquire.

**Recommendation:** Do not set the value to 0, because it can cause the IRLM to experience storage shortages.

**Related tasks:**
- Specifying a default value for the LOCKMAX option (Db2 Performance)

**Related reference:**
- LOCKS PER USER field (NUMLKUS subsystem parameter)
- CREATE TABLESPACE (Db2 SQL)
- ALTER TABLESPACE (Db2 SQL)

**LOCKS PER USER field (NUMLKUS subsystem parameter)**
The NUMLKUS subsystem parameter specifies the maximum number of page, row, or LOB locks that a single application can hold concurrently for all table spaces.

---

**Acceptable values:** 0 - 104857600
**Default:** 10000
**Update:** option 33 on panel DSNTIPB
**DSNZPxxx:** DSN6SPRM NUMLKUS

---

**0 - 104857600**

The maximum number includes locks on data pages, LOBs, XMLs, and rows that the program acquires when it accesses table spaces. The limit applies to all table spaces that are defined with the LOCKSIZE PAGE, LOCKSIZE ROW, or LOCKSIZE ANY options. A value of 0 indicates that there is no limit to the number of data and row locks that a program can acquire.

You can enter the number of locks as an integer or you can enter a value with a suffix of K or M. If you enter a value with a suffix of K, the number of locks is that value multiplied by 1024. If you enter a value with a suffix of M, the number of locks is that value multiplied by 1,048,576.

Db2 assumes that each lock requires 540 bytes of storage. If you define referential constraints between values, you might want to select a higher value for this field.

To avoid exhausting the IRLM's storage for locks, follow these guidelines:
- Do not specify 0 or a very large value unless it is specifically required to run an application.
- Consider the design of your applications. Long-running applications, particularly those that perform row-level locking, have few or infrequent commit points, or...
use repeatable-read isolation may use substantial amounts of lock storage. You should perform frequent commits to release locks.

**Important:** These values are constraints for a single application. Each concurrent application can hold the maximum number of locks specified here.

Check panel DSNTIPC to ensure that the required storage for the IRLM does not exceed the available region size for the IRLM.

**Related tasks:**

- [Specifying the maximum number of locks that a single process can hold (Db2 Performance)](Db2 Performance)

**DEADLOCK TIME field**
The DEADLOCK TIME field controls the amount of time for which local deadlock detection cycles are to run. A deadlock is a situation in which two or more requesters are waiting for resources that are held by another requester. Deadlock detection is the procedure that identifies deadlocks and its participants.

Acceptable values: 1 - 5, 100 to 5000
Default: 1
Update: edit IRLM start procedure
DSNZPxxx: none

Db2 interprets values between 1 and 5 as seconds and values between 100 and 5000 as milliseconds. Depending on the value that you enter, IRLM might substitute a smaller maximum value.

The associated IRLM PROC parameter is DEADLOK.

**DEADLOCK CYCLE field**
The value of the DEADLOCK CYCLE field specifies the number of local deadlock cycles that must expire before the IRLM does global deadlock detection processing.

Acceptable values: 1
Default: 1
Update: edit IRLM start procedure
DSNZPxxx: none

The DEADLOCK CYCLE field is used only for Db2 data sharing.

The associated IRLM PROC parameter is DEADLOK.

**MEMBER IDENTIFIER field**
The value of the MEMBER IDENTIFIER field specifies an ID number that uniquely names this IRLM member within an IRLM data sharing group.

Acceptable values: 1 - 255
Default: 1
Update: edit IRLM start procedure
DSNZPxxx: none

**Recommendation:** Correlate the IRLM member ID with the Db2 member name. For example, for Db2 member DSN1, specify an IRLM member ID of 1.
The IRLM ID that you specify does not relate directly to the limit of IRLM members that can be in the data sharing group. That limit is determined by the current hardware limits (32). If you edit the IRLMPROC directly, you can specify a value from 1 - 255.

This option is used only for Db2 data sharing.

The associated IRLM PROC parameter is IRLMID.

**Related information:**

[About Db2 and related commands (Db2 Commands)](#)

### IRLM XCF GROUP NAME field

The value of the IRLM XCF GROUP NAME field is the name that is to be used for the IRLM group. This name must be different from the Db2 group name.

- **Acceptable values:** 1 - 8 characters
- **Default:** DXRGROUP
- **Update:** edit IRLM start procedure
- **DSNZPxxx:** none

**Recommendation:** Begin this name with DXR. All members in the Db2 group must have the same IRLM XCF group name.

This option is used only for Db2 data sharing.

To avoid names that IBM uses for its cross-system coupling facility (XCF) groups, the first character must be an uppercase letter J - Z unless the name begins with DXR. Do not use SYS as the first three characters, and do not use UNDESIG as the group name.

The associated IRLM PROC parameter is IRLMGRP

### LOCK ENTRY SIZE field

The value of the LOCK ENTRY SIZE field specifies the initial size, in bytes, of individual lock entries in the lock table portion of the lock structure.

- **Acceptable values:** 2, 4, 8
- **Default:** 2
- **Update:** edit IRLM start procedure
- **DSNZPxxx:** none

**Recommendation:** If you have seven or fewer members in your data sharing group, use the default value for the most efficient use of coupling lock structure space. If you have more than seven members, use a value of 4 to avoid an automatic rebuild for maximum connections.

Db2 converts the value for LOCK ENTRY SIZE to a corresponding value for the IRLM parameter MAXUSRS as shown in the table below.

**Table 31. Converting lock entry size to MAXUSRS values**

<table>
<thead>
<tr>
<th>LOCK ENTRY SIZE</th>
<th>MAXUSRS value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
</tr>
</tbody>
</table>
NUMBER OF LOCK ENTRIES field

The value of the NUMBER OF LOCK ENTRIES field specifies the number of lock table entries that are to be in the coupling facility lock structure.

Acceptable values: 1 - 2048
Default: 0
Update: edit IRLM start procedure
DSNZPxxx: none

This value must be a power of 2 in the range of 1 - 2048, with each increment representing 1,048,576 lock table entries. The default value, 0, indicates that IRLM is to determine the number of lock table entries based on the lock structure size that is specified in CFRM policy and the number of users (MAXUSRS). If you specify a value for lock structure size in CFRM policy that is greater than 2048 MB, IRLM limits the number of lock table entries to a maximum of 2048 MB. If you want to control the number of lock table entries, enter a non-zero value within the accepted range.

The number of lock table entries has a direct effect on cross-system extended services (XES) contention. You should therefore monitor this number to find the optimum values for your installation.

The associated IRLM PROC parameter is LTE.

DISCONNECT IRLM field

The value of the DISCONNECT IRLM field determines whether IRLM is to automatically disconnect from the data sharing group when Db2 is not identified to it.

Acceptable values: YES, NO
Default: YES
Update: edit IRLM start procedure
DSNZPxxx: none

YES
Causes IRLM to disconnect from the data sharing group when Db2 is stopped normally or it stops as the result of a Db2 failure.

If you specify YES for the DISCONNECT IRLM field and YES for the AUTO START field on panel DSNTIPI, stopping IRLM does not require manual intervention.

NO IRLM remains connected to the data sharing group even when Db2 is stopped. In this case, you must explicitly stop IRLM to bring it down.

This option has less impact on other systems when a Db2 fails because z/OS is not required to perform certain recovery actions that it normally performs when IRLM comes down. Specifying NO can also mean that Db2 restarts more quickly after a Db2 normal or abnormal termination because it does not have to wait for IRLM to rejoin the IRLM data sharing group.

The associated IRLM PROC parameter is SCOPE.
The fields on the DSNTIP6 panel configure the default behavior of enhancements to the BACKUP SYSTEM, RESTORE SYSTEM, RECOVER, and other utilities.

<table>
<thead>
<tr>
<th>DSNTIP6</th>
<th>UPDATE DB2 - DB2 UTILITIES PARAMETERS 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>=&gt;</code></td>
<td><strong>TEMP DS UNIT NAME</strong> ====&gt; <strong>SYSDA</strong></td>
</tr>
<tr>
<td><strong>1</strong></td>
<td>Device for temporary utility data sets</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>UTIL TEMP STORCLAS</strong> ====&gt; <strong>NONE</strong></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>For temporary shadow data sets by utilities</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>STATISTICS HISTORY</strong> ====&gt; <strong>NONE</strong></td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Default for collection of statistics history</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>STATISTICS ROLLUP</strong> ====&gt; <strong>YES</strong></td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Allow statistics aggregation: NO, YES</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>UTILITY TIMEOUT</strong> ====&gt; <strong>6</strong></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>Utility wait time multiplier</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>IGNORE SORTNUM STMT</strong> ====&gt; <strong>NO</strong></td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>Ignore SORTNUM keyword in UT stmt</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>TEMPLATE TIME</strong> ====&gt; <strong>UTC</strong></td>
</tr>
<tr>
<td><strong>7</strong></td>
<td>For TEMPLATE TIME default: UTC, LOCAL</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>MAX UTILS PARALLELISM</strong> ====&gt; <strong>99</strong></td>
</tr>
<tr>
<td><strong>8</strong></td>
<td>Maximum degree of parallelism for Utilities: 0 - 32767</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>STATISTICS COLGROUP</strong> ====&gt; <strong>10</strong></td>
</tr>
<tr>
<td><strong>9</strong></td>
<td>Storage limit in MB for sorting single column COLGROUPs: 0 - 4096</td>
</tr>
<tr>
<td><code>=&gt;</code></td>
<td><strong>DATA SORT STG LIMIT</strong> ====&gt; <strong>SYSTEM</strong></td>
</tr>
<tr>
<td><strong>10</strong></td>
<td>Collect RUNSTATS using page-level sampling by default (YES, NO, SYSTEM)</td>
</tr>
</tbody>
</table>

**TEMP DS UNIT NAME field (VOLTDEVT subsystem parameter)**

The VOLTDEVT subsystem parameter specifies the device type or unit name that is to be used by the Db2 COPY CONCURRENT and CHECK utilities for dynamically allocating temporary data sets. Specifying a value for VOLTDEVT removes the need to specify the FILTERDDN DD in the utility job.

Acceptable values: valid device type or unit name
Default: SYSDA
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM VOLTDEVT

The temporary data sets are deleted when the utility execution is complete.

**UTIL TEMP STORCLAS field (UTIL_TEMP_STORCLAS subsystem parameter)**

The UTIL_TEMP_STORCLAS subsystem parameter specifies the storage class that the CHECK INDEX, CHECK DATA, and CHECK LOB utilities are to use when allocating temporary shadow data sets. (These utilities allocate shadow data sets when you specify the SHRLEVEL CHANGE option.)

Acceptable values: blank or a valid SMS management class name
Default: blank
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM UTIL_TEMP_STORCLAS

The default value of blank indicates that the shadow data sets are to be defined in the same storage class as the production page set.
Recommendation: Specifying an explicit storage class for this subsystem parameter can improve performance, especially when the system uses paired disk volumes for peer to peer remote copy.

In a data sharing group, this parameter has group scope.

**STATISTICS HISTORY field (STATHIST subsystem parameter)**
The STATHIST subsystem parameter specifies which inserts and updates are to be recorded in catalog history tables.

Acceptable values: SPACE, NONE, ALL, ACCESSPATH  
Default: NONE  
Update: option 34 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM STATHIST

**SPACE**  
All inserts and updates that Db2 makes to space-related catalog statistics are recorded.

**NONE**  
None of the changes that Db2 makes in the catalog are recorded. NONE is the default value.

**ALL**  
All inserts and updates that Db2 makes in the catalog are recorded.

**ACCESSPATH**  
All inserts and updates that Db2 makes to ACCESSPATH-related catalog statistics are recorded.

Related tasks:
- [Collecting history statistics (Db2 Performance)](#)  
- [Collecting statistics history (Db2 Utilities)](#)

Related reference:
- [History statistics (Db2 Performance)](#)

**STATISTICS ROLLUP field (STATROLL subsystem parameter)**
The STATROLL subsystem parameter specifies whether the RUNSTATS utility is to aggregate the partition-level statistics, even though some parts might not contain data.

Acceptable values: YES, NO  
Default: YES  
Update: option 34 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM STATROLL

For Db2 subsystems that have large partitioned table spaces and indexes, it is recommended that you specify YES for STATISTICS ROLLUP. Specifying YES for STATISTICS ROLLUP enables the aggregation of partition-level statistics and helps the optimizer to choose a better access path.

**UTILITY TIMEOUT field (UTIMOUT subsystem parameter)**
The UTIMOUT subsystem parameter specifies how long, in number of resource values, that a utility or utility command is to wait for a resource. The utility or utility command waits until a lock or all claims on a resource of a particular claim class is released.
Acceptable values: 1 - 254
Default: 6
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM UTIMOUT

For example, if you use the default value of six, a utility can wait six times longer than an SQL application for a resource. This option allows utilities to wait longer than SQL applications to access a resource. For more information about setting the resource timeout for SQL applications, see the discussion of the RESOURCE TIMEOUT field on DSNTIP.

The value of UTILITY TIMEOUT is used as the default value for the RETRY parameter of Db2 Utilities such as CHECK INDEX and online REBUILD INDEX.

Related tasks:

- Specifying the amount of inactive time before a timeout (Db2 Performance)

Related reference:

- RESOURCE TIMEOUT field (IRLMRWT subsystem parameter)

**IGNORE SORTNUM STMT field (IGNSORTN subsystem parameter)**

The IGNSORTN subsystem parameter determines whether occurrences of the SORTNUM clause in utility control statements are to be ignored.

Acceptable values: YES or NO
Default: NO
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM IGNSORTN

**YES**

Db2 ignores any SORTNUM specifications in utility control statements, and utilities attempt to allocate the sort work data sets.

**NO**

Any SORTNUM specifications in utility control statements are passed to the sort program. The SORTNUM specification is used to determine the number of sort work data sets to dynamically allocate.

**TEMPLATE TIME field (TEMPLATE_TIME subsystem parameter)**

The TEMPLATE_TIME subsystem parameter specifies the default setting for the TIME option of the TEMPLATE statement.

Tip: Set all Db2 data sharing members to the same value.

Acceptable values: UTC, LOCAL
Default: UTC
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM TEMPLATE_TIME

**UTC**

Coordinated Universal Time.

**LOCAL**

Local time at the Db2 database manager.

Related reference:

- Syntax and options of the TEMPLATE control statement (Db2 Utilities)
MAX UTILS PARALLELISM field (PARAMDEG_UTIL subsystem parameter)
The PARAMDEG_UTIL subsystem parameter specifies the maximum number of parallel subtasks for some utilities.

PARAMDEG_UTIL affects the following utilities:
- REORG TABLESPACE
- REBUILD INDEX
- CHECK INDEX
- UNLOAD
- LOAD
- COPY
- RECOVER

Acceptable values: 0 - 32767
Default: 99
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM PARAMDEG_UTIL

0  No additional constraint is placed on the maximum degree of parallelism in a utility.
1 - 32767  Specifies the maximum number of parallel subtasks for all affected utilities.

STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter)
The STATCLGSRT subsystem parameter specifies how the sort program is to be used by the RUNSTATS utility or by an inline statistics job when collecting statistics on a single column that is defined with the COLGROUP option.

Acceptable values: 0 - 4096 MB
Default: 10
Update: option 34 on panel DSNTIPB
DSNZPxxx: DSN6SPRM STATCLGSRT
Data sharing scope: Member
Online changeable: Yes

When you specify FREQVAL to collect statistics on single-column COLGROUPs, consider adjusting the value of the STATCLGSRT subsystem parameter to avoid sort, if both of the following conditions exist:
- The specified COUNT value is less than or equal to 1000.
- The LEAST and BOTH options are not specified.

If either of these conditions does not exist or if STATCLGSRT specifies 0, Db2 does a data sort, which can negatively affect the performance of statistics collection. A non-zero value for STATCLGSRT indicates the amount of memory space, specified in MB, that the utility can use to avoid the data sort. Specifying a non-zero value is likely to improve performance and reduce costs for collecting statistics in these circumstances, but storage costs might increase.

0  An indication that a data sort is always performed.
The amount of memory space for the utility to use to avoid the data sort. Values are specified in megabytes (for example, 4096, which is the maximum allowable value). Increasing the value might be necessary when utility jobs specify a large number of single-column COLGROUPs, and message DSNU1388I is issued.

Although histogram statistics can be collected in the same RUNSTATS or inline statistics job, any COLGROUPs for which histogram statistics are being collected are not eligible for the sort-avoidance performance improvement. You can override the STATCLGSMRT value for a particular utility job by specifying the STATCLGMEMSRRT option on the utility statement.

Related tasks:
- [Reducing the cost of collecting statistics (Db2 Performance)](#)
- [RUNSTATS TABLESPACE syntax and options (Db2 Utilities)](#)

**PAGE-LEVEL SAMPLING field (STATPGSAMP subsystem parameter)**

The STATPGSAMP subsystem parameter specifies whether the RUNSTATS utility uses page-level sampling by default for universal table spaces. STATPGSAMP applies to only universal table spaces, and has no effect on other types of table spaces. Also, STATPGSAMP applies only when the TABLE option is specified in the RUNSTATS statement.

Acceptable values: YES, NO, SYSTEM
Default: SYSTEM
Update: option 34 on panel DSNTIPB
DSNZP: DSN6SPRM STATPGSAMP
Data sharing scope: Member
Online changeable: Yes

**SYSTEM**

Db2 determines the default sampling method.

Currently, SYSTEM has the same meaning as NO. In a future function level, SYSTEM might change to mean YES (use TABLESAMPLE SYSTEM AUTO by default). Any such change will be included in the documentation for that function level.

**YES**

RUNSTATS always executes as if the TABLESAMPLE SYSTEM AUTO utility option is specified. (TABLESAMPLE SYSTEM AUTO specifies that RUNSTATS uses page sampling, and the utility determines the sampling rate.) The only exception is if the RUNSTATS statement contains TABLESAMPLE SYSTEM and a value other than AUTO. In that case, RUNSTATS uses that value instead of AUTO.

If the SAMPLE utility option is specified to request row sampling, RUNSTATS ignores that option and uses page sampling instead.

**NO**

RUNSTATS does not use page sampling by default. Any sampling is determined by the options that are specified in the RUNSTATS statement.

Related tasks:
DSNTIP61: Db2 utilities parameters panel 2

The fields on the DSNTIP61 panel are used to configure the default behavior of various Db2 utilities.

COPY FAST REPLICATION field (COPY_FASTREPLICATION subsystem parameter)

Specify for the COPY utility whether FlashCopy fast replication is preferred, required, or not used for the creation of the FlashCopy image copy.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: PREFERRED, REQUIRED, NONE
Default: PREFERRED
Data sharing scope: It is recommended that all members use the same setting.
Update: option 35 on panel DSNTIPB
Online changeable: Yes
DSNZPxxx: DSN6SPRM.COPY_FASTREPLICATION

PREFERRED
The COPY utility directs DSS COPY to use fast replication if possible. If FlashCopy® cannot be used, then DSS uses traditional data movement methods.

REQUIRED
The COPY utility directs DSS COPY to only use fast replication, ensuring that object copies occur as quickly as possible. This option causes the COPY utility to fail if FlashCopy cannot be used. This option might reduce the opportunity for resource contention and unavailability for SHRLEVEL REFERENCE copies.

NONE
The COPY utility directs DSS COPY not to use fast replication. Traditional data movement methods are be used.
Related concepts:

**Subsystem parameters for refining DFSMSdss COPY operation with utilities (Db2 Utilities)**

**FAST RESTORE field (REC_FASTREPLICATION subsystem parameter)**
The REC_FASTREPLICATION parameter specifies whether the RECOVER utility should use FlashCopy to recover from a FlashCopy image copy.

- **Acceptable values:** NONE, PREFERRED, REQUIRED
- **Default:** PREFERRED
- **Update:** option 35 on panel DSNTIPB
- **DSNZPxxx:** DSN6SPRM.REC_FASTREPLICATION

**NONE**
The RECOVER utility uses standard I/O to restore a FlashCopy image copy.

**PREFERRED**
The RECOVER utility uses FlashCopy to recover from a FlashCopy image copy if FlashCopy support is available.

**REQUIRED**
The RECOVER utility forces the use of FlashCopy to recover from a FlashCopy image copy to ensure that recovery occurs as quickly as possible. This option causes RECOVERY to fail if FlashCopy cannot be used.

If you use BACKUP SYSTEM to create system-level backups, using FlashCopy for a restore might cause BACKUP SYSTEM to fail because bi-directional FlashCopy is not supported.

**Related concepts:**

**Subsystem parameters for refining DFSMSdss COPY operation with utilities (Db2 Utilities)**

**FLASHCOPY PPRC field (FLASHCOPY_PPRC subsystem parameter)**
The FLASHCOPY_PPRC subsystem parameter specifies the behavior for DFSMSdss FlashCopy requests when the target disk storage volume is the primary device in a peer-to-peer remote copy (Metro Mirror) relationship.

Specifically, FLASHCOPY_PPRC determines the following:
- Whether DFSMSdss preserves mirroring while processing a Db2 utilities request
- Whether the target device pair is allowed to go to duplex pending state

**Important:** Set this field to blank if PPRC is not available on the subsystem.

- **Acceptable values:** NONE, PREFERRED, REQUIRED, blank
- **Default:** REQUIRED
- **Update:** option 35 on panel DSNTIPB
- **DSNZPxxx:** DSN6SPRM.FLASHCOPY_PPRC

**NONE**
DFSMSdss does not preserve mirroring while processing a Db2 utilities request, even if all the configuration requirements for preserving the mirror operation are met.
DFSMSdss allows a volume pair to go into the duplex pending state while processing a Db2 utilities request only if it is required.

The IBM DS8880 series and newer IBM storage systems do not support the Preserve Mirror Preferred option for PPRC. If you are using one of those storage systems, and you want to preserve your disk mirror or copy, specify REQUIRED.

DFSMSdss does not permit a PPRC volume pair to go into duplex pending state while processing a Db2 utilities request.

No preserve mirror command is passed to DFSMSdss.

## Related concepts:

- **Subsystem parameters for refining DFSMSdss COPY operation with utilities (Db2 Utilities)**
- **FLASHCOPY XRCP field (FLASHCOPY_XRCP subsystem parameter)**

### FLASHCOPY XRCP field (FLASHCOPY_XRCP subsystem parameter)

The FLASHCOPY_XRCP subsystem parameter specifies whether a data set level FlashCopy operation is allowed to a primary volume in an XRC relationship when the z/OS DFSMSdss support for RPFC for XRC is installed and enabled.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>YES, NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>YES</td>
</tr>
<tr>
<td>Update</td>
<td>option 35 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZP.xxxx:</td>
<td>DSN6SPRM FLASHCOPY_XRCP</td>
</tr>
</tbody>
</table>

**YES** If z/OS DFSMSdss support for RPFC in XRC is installed and enabled, supported utilities specify the FCTOXRCPrimary option when they issue DFSMSdss COPY commands.

YES is ignored in the following situations:

- For CHECK utilities, if the CHECK_FASTREPLICATION subsystem parameter value is NONE.
- For the RECOVER utility, if the REC_FASTREPLICATION subsystem parameter value is NONE.
- For the COPY utility, if the COPY_FASTREPLICATION subsystem parameter value is NONE.

**NO** The utilities do not specify the FCTOXRCPrimary option when they issue DFSMSdss COPY commands.

## Related concepts:

- **Subsystem parameters for refining DFSMSdss COPY operation with utilities (Db2 Utilities)**
- **Db2 online utilities (Db2 Utilities)**

## DEFAULT TEMPLATE field (FCCOPYDDN subsystem parameter)

The FCCOPYDDN subsystem parameter defines the default value that is to be used for the FCCOPYDDN parameter of the FLASHCOPY option of Db2 utilities.
control statements. This parameter applies to the COPY, LOAD, REBUILD INDEX, REORG INDEX, and REORG TABLESPACE utilities.

FCCOPYDDN specifies the template for deriving the data set name for the FlashCopy image copy.

FlashCopy technology can be used to create VSAM copies of objects almost instantaneously. It creates a separate data set for each piece of the object that is being copied. FlashCopy technology can also be used to recover the object if necessary, which reduces recovery time and therefore provides greater availability.

**Important:** You must replace “HLQ” in the default value with a high-level qualifier that is appropriate for your subsystem.

Acceptable values: valid Db2 utilities template
Default: HLQ.&DB..&SN..N&DSNUM..&UQ.
Update: option 35 on panel DSNTIPB
DSNZPxxx: DSN6SPRM FCCOPYDDN

If you assign the FCCOPYDDN setting directly in DSNTIJUZ, you need to repeat each ampersand (&) character to conform with IBM High-Level Assembler rules. For example, you would need to assign the default value above as FCCOPYDDN=HLQ.&DB..&SN..N&DSNUM..&UQ. (notice the trailing period).

**COPY field (FLASHCOPY_COPY subsystem parameter)**
The value of the FLASHCOPY_COPY parameter specifies whether the FLASHCOPY option of the COPY utility is to be used by default.

Acceptable values: NO, YES
Default: NO
Update: option 35 on panel DSNTIPB
DSNZPxxx: DSN6SPRM FLASHCOPY_COPY

- **NO** Specifies that the FLASHCOPY option of the COPY utility is not used by default.
- **YES** Specifies that the COPY utility uses FlashCopy technology when the FLASHCOPY option is not specified in the control statement. Specify YES only if the Db2 data sets are on FlashCopy Version 2 DASD.

**LOAD field (FLASHCOPY_LOAD subsystem parameter)**
The value of the FLASHCOPY_LOAD subsystem parameter specifies whether the FLASHCOPY option of the LOAD utility is to be used by default.

Acceptable values: NO, YES
Default: NO
Update: option 35 on panel DSNTIPB
DSNZPxxx: DSN6SPRM FLASHCOPY_LOAD

- **NO** Specifies that the FLASHCOPY option of the LOAD utility is not used by default.
- **YES** Specifies that the LOAD utility uses FlashCopy technology when the
FLASHCOPY option is not specified in the control statement. Specify YES only if the Db2 data sets are on FlashCopy Version 2 DASD.

REORG TABLESPACE field (FLASHCOPY_REORG_TS subsystem parameter)
The value of the FLASHCOPY_REORG_TS subsystem parameter specifies whether the FLASHCOPY option of the REORG TABLESPACE utility is to be used by default.

Acceptable values: NO, YES
Default: NO
Update: option 35 on panel DSNTIPB
DSNZPxxx: DSN6SPRM FLASHCOPY_REORG_TS

NO Specifies that the FLASHCOPY option of the REORG TABLESPACE utility is not used by default.

YES Specifies that the REORG TABLESPACE utility uses FlashCopy technology when the FLASHCOPY option is not specified in the control statement. Specify YES only if the Db2 data sets are on FlashCopy Version 2 DASD.

REBUILD INDEX field (FLASHCOPY_REBUILD_INDEX subsystem parameter)
The value of the FLASHCOPY_REBUILD_INDEX subsystem parameter specifies whether the FLASHCOPY option of the REBUILD INDEX utility is to be used by default.

Acceptable values: NO, YES
Default: NO
Update: option 35 on panel DSNTIPB
DSNZPxxx: DSN6SPRM FLASHCOPY_REBUILD_INDEX

NO Specifies that the FLASHCOPY option of the REBUILD INDEX utility is not used by default.

YES Specifies that the REBUILD INDEX utility uses FlashCopy technology when the FLASHCOPY option is not specified in the control statement. Specify YES only if the Db2 data sets are on FlashCopy Version 2 DASD.

REORG INDEX field (FLASHCOPY_REORG_INDEX subsystem parameter)
The value of the FLASHCOPY_REORG_INDEX subsystem parameter specifies whether the FLASHCOPY option of the REORG INDEX utility is to be used by default.

Acceptable values: NO, YES
Default: NO
Update: option 35 on panel DSNTIPB
DSNZPxxx: DSN6SPRM FLASHCOPY_REORG_INDEX

NO Specifies that the FLASHCOPY option of the REORG INDEX utility is not used by default.

YES Specifies that the REORG INDEX utility uses FlashCopy technology when the
FLASHCOPY option is not specified in the control statement. Specify YES only if the Db2 data sets are on FlashCopy Version 2 DASD.

**DSNTIP62: Db2 utilities parameters panel 3**

The fields on the DSNTIP62 panel are used to configure the default behavior of various Db2 utilities.

---

**SYSTEM-LEVEL BACKUPS field (SYSTEM_LEVEL_BACKUPS subsystem parameter)**

The RECOVER utility uses image copies and concurrent copies for object-level recoveries. The SYSTEM_LEVEL_BACKUPS subsystem parameter specifies whether the RECOVER utility is to also use system-level backups as a recovery base for object-level recoveries.

| Acceptable values: | NO, YES |
| Default: | NO |
| Update: | option 36 on panel DSNTIPB |
| DSNZPxxx: | DSN6SPRM SYSTEM_LEVEL_BACKUPS |

Specify the default of NO if you do not take system-level backups with the BACKUP SYSTEM utility. Specify YES if you take system-level backups with the BACKUP SYSTEM utility.

**Related reference:**

[RECOVER (Db2 Utilities)]

**RESTORE/RECOVER field (RESTORE_RECOVER_FROMDUMP subsystem parameter)**

The RESTORE_RECOVER_FROMDUMP subsystem parameter controls whether the system-level backup for the RESTORE SYSTEM and the RECOVER utilities is to occur from the dump on tape.

| Acceptable values: | YES or NO |
| Default: | NO |
| Update: | option 36 on panel DSNTIPB |
| DSNZPxxx: | DSN6SPRM RESTORE_RECOVER_FROMDUMP |

---

Figure 41. DB2 utilities parameters panel 3: DSNTIP62
YES
System-level backup for the RESTORE SYSTEM and the RECOVER utilities occurs from the dump on tape.

NO
System-level backup for the RESTORE SYSTEM and the RECOVER utilities occurs from the disk copy of the system-level backup.

You can override the setting of RESTORE/RECOVER by executing the RESTORE SYSTEM utility statement or the RECOVER utility statement with the FROMDUMP keyword.

DUMP CLASS NAME field (UTILS_DUMP_CLASS_NAME subsystem parameter)
A system-level backup can occur from a dump on tape or from a disk copy. For backups that occur from the dump on tape, the UTILS_DUMP_CLASS_NAME subsystem parameter specifies the name of the DFSMSHsm dump class that is to be used.

Acceptable values: blank or a valid DFSMSHsm dump class name not exceeding 8 characters in length
Default: blank
Update: option 36 on panel DSNTIPB
DSNZPxxx: DSN6SPRM UTILS_DUMP_CLASS_NAME

This dump class is to be used by the RECOVER utility and the RESTORE SYSTEM utility to restore objects from a system-level backup that has been dumped to tape. The setting for this field is applicable only when you specify YES in the RESTORE/RECOVER field on panel DSNTIP6. You can override the setting for DUMP CLASS NAME by executing the RESTORE SYSTEM utility statement or the RECOVER utility statement with the DUMPCLASS keyword.

Related concepts:
[Dump Tasks (z/OS DFSMSHsm Storage Administration)]

MAXIMUM TAPE UNITS field (RESTORE_TAPEUNITS subsystem parameter)
When the RESTORE SYSTEM utility restores a system-level backup that has been dumped to tape, it allocates a number of tape units or tape drives. The RESTORE_TAPEUNITS subsystem parameter specifies the maximum number of tape units or tape drives that the RESTORE SYSTEM utility can allocate.

Acceptable values: NOLIMIT or 1 - 255
Default: NOLIMIT
Update: option 36 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RESTORE_TAPEUNITS

The default value, NOLIMIT, means that the RESTORE SYSTEM utility will allocate as many tape units as necessary to restore the system-level backup. You can override the setting for MAXIMUM TAPE UNITS by executing the RESTORE SYSTEM utility statement with the TAPEUNITS keyword.

ALTERNATE COPY POOL field (ALTERNATE_CP subsystem parameter)
The ALTERNATE_CP subsystem parameter specifies the name of an alternate SMS copy pool for the BACKUP SYSTEM utility. If a copy pool is specified, BACKUP
SYSTEM alternates between using that copy pool and the standard copy pool when the utility creates system-level backups.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: blank or 1-14 alphanumeric characters, including @, #, and $. The first character must be non-numeric. Db2 uses this value to build the name according to the Db2 naming convention for copy pools.

Default: blank
Update: option 36 on panel DSNTIPB
DSNZPRMxxx: DSN6SPRM ALTERNATE_CP
Data sharing scope: All members should use the same setting.

The value of the ALTERNATE_CP subsystem parameter applies only if the ALTERNATE_CP option is not specified in the BACKUP SYSTEM utility control statement. If the ALTERNATE_CP option is specified in the BACKUP SYSTEM statement, that value overrides the subsystem parameter value. In that case, the copy pool that is specified in the BACKUP SYSTEM statement is used as the alternate copy pool for that utility invocation only.

The value of the ALTERNATE_CP subsystem parameter has the following meanings:

- **blank**
  BACKUP SYSTEM uses the standard copy pool only.

- **Not blank**
  BACKUP SYSTEM uses the specified copy pool as the alternate copy pool for system-level backups. BACKUP SYSTEM alternates between using the standard copy pool and the alternate copy pool for system-level backups. BACKUP SYSTEM checks the BSDS to determine whether the standard copy pool or alternate copy pool was used for the most recent system-level backup. If no system-level backups are recorded in the BSDS, BACKUP SYSTEM uses the standard copy pool. If system-level backups are recorded in the BSDS, BACKUP SYSTEM uses the copy pool that was not used for the most recent system-level backup.

BACKUP SYSTEM, RESTORE SYSTEM, and RECOVER can create and use system-level backups for the standard copy pool and more than one alternate copy pool. However, only the standard copy pool and the alternate copy pool that is specified in the ALTERNATE_CP subsystem parameter are used for automatically alternating the copy pool for system-level backups. Other copy pool names are ignored.

The ALTERNATE_CP value applies to both the database copy pool and the log copy pool, if one is specified.

**Related concepts:**

- Copy pools (Db2 Utilities)

**Related reference:**

- Syntax and options of the BACKUP SYSTEM control statement (Db2 Utilities)
DB BACKUP STG GROUP field (UTIL_DBBSG subsystem parameter)
The UTIL_DBBSG subsystem parameter specifies the name of a backup SMS
storage group to be used by the BACKUP SYSTEM utility for the database copy
pool. This field is optional. You can set this field to a non-blank value only if the
value of the ALTERNATE COPY POOL field is also non-blank.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated
function level is function level 500 or higher.

Acceptable values: blank or a valid SMS storage group name
Default: blank
Update: option 32 on panel DSNTPPB
DSNZPxxx: DSN6SPRM UTIL_DBBSG
Data sharing scope: All members should use the same setting.

blank
BACKUP SYSTEM is to request that DFSMShsm use the copy pool backup
storage group that is associated with each storage group in the copy pool.

Valid SMS storage group name
BACKUP SYSTEM is to request that DFSMShsm use the specified backup
storage group. The specified backup storage group must be defined with the
COPY POOL BACKUP attribute.

Related concepts:
Copy pools (Db2 Utilities)

Related reference:
ALTERNATE COPY POOL field (ALTERNATE_CP subsystem parameter)
Syntax and options of the BACKUP SYSTEM control statement (Db2 Utilities)

LOG BACKUP STG GRP field (UTIL_LGBSG subsystem parameter)
The UTIL_LGBSG subsystem parameter specifies the name of a backup SMS
storage group to be used by the BACKUP SYSTEM utility for the log copy pool.
This field is optional. You can set this field to a non-blank value only if the value
of the ALTERNATE COPY POOL field is also non-blank.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated
function level is function level 500 or higher.

Acceptable values: blank or a valid SMS storage group name
Default: blank
Update: option 36 on panel DSNTPPB
DSNZPxxx: DSN6SPRM UTIL_LGBSG
Data sharing scope: All members should use the same setting.

blank
BACKUP SYSTEM is to request that DFSMShsm use the copy pool backup
storage group that is associated with each storage group in the copy pool.

Valid SMS storage group name
BACKUP SYSTEM is to request that DFSMShsm use the specified backup
storage group. The specified backup storage group must be defined with the
COPY POOL BACKUP attribute.

Related concepts:
HSM MESSAGE DS HLQ field (UTILS_HSM_MSGDS_HLQ subsystem parameter)

Specifies the high-level qualifier for data sets to be allocated by the Db2 BACKUP SYSTEM and RESTORE SYSTEM utilities in order to receive messages from IBM Hierarchical Storage Management (HSM) and IBM Data Facility Data Set Services (DFDSS). These messages will be included for diagnostic purposes in Db2 utility SYSPRINT DD output.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: Valid data set qualifier of 1 - 6 characters
Default: blank
Update: option 36 on panel DSNTIPB
Online changeable: Yes
DSNZPxxx: DSN6SPRM.UTILS_HSM_MSGDS_HLQ
Data sharing scope: All members should use the same setting.

This high-level qualifier must also be registered in HSM via a SETSYS command. Data sets that use this high-level qualifier are defined and populated by HSM and DFDSS during BACKUP SYSTEM and RECOVER SYSTEM processing, then allocated by Db2 and the content written to utility's SYSPRINT DD. Db2 does not delete the data set. These data sets do not include messages from DUMP processing because control is returned to Db2 before dump processing is complete.

DSNTIP63: Db2 utilities parameters panel 4

The fields on the DSNTIP63 panel are used to configure the default behavior of various Db2 utilities.

Figure 42. DB2 utilities parameters panel 3: DSNTIP63

REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter)

The REORG_PART_SORT_NPSI subsystem parameter specifies whether the REORG TABLESPACE PART utility decides to sort all of the keys of a
non-partitioned secondary index. The setting is ignored for a REORG that is not part-level or without non-partitioned secondary indexes.

Acceptable values: AUTO, NO, YES
Default: AUTO
Update: option 37 on panel DSNTIPB
DSNZPxxx: DSN6SPRM REORG_PART_SORT_NPSI

AUTO
Specifies that if sorting all keys of the non-partitioned secondary indexes improves the elapsed time and CPU performance, all keys are sorted.

YES
Specifies that if sorting all keys of the non-partitioned secondary indexes improves the elapsed time, all keys are sorted.

NO
Specifies that only keys of the non-partitioned secondary indexes that are in the scope of the REORG are sorted.

REORG LIST PROCESSING field (REORG_LIST_PROCESSING subsystem parameter)
The REORG_LIST_PROCESSING subsystem parameter specifies the default setting for the PARALLEL option of the Db2 REORG TABLESPACE utility.

Acceptable values: PARALLEL, SERIAL
Default: PARALLEL
Update: option 37 on panel DSNTIPB
DSNZPxxx: DSN6SPRM REORG_LIST_PROCESSING

PARALLEL
The default value PARALLEL specifies that the REORG TABLESPACE utility is to use a default PARALLEL YES option when the PARALLEL keyword is not specified in the utility control statement. The PARALLEL YES option specifies that the REORG TABLESPACE utility is to process all partitions that are specified in the input LISTDEF statement in a single execution of the utility.

SERIAL
SERIAL specifies that the REORG TABLESPACE utility is to use a default PARALLEL NO option when the PARALLEL keyword is not specified in the utility control statement. The PARALLEL NO option specifies that each partition that is specified in the input LISTDEF statement is to be processed in a separate execution of the utility.

REORG MAPPING DB field (REORG_MAPPING_DATABASE subsystem parameter)
The REORG_MAPPING_DATABASE subsystem parameter specifies the default database that REORG TABLESPACE SHRLEVEL CHANGE uses to implicitly create the mapping table.

Acceptable values: database-name
Default: blank
Update: option 37 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.REORG_MAPPING_DATABASE
Online changeable: Yes
Data sharing scope: Member

database-name
The name of the default database that REORG TABLESPACE SHRLEVEL
CHANGE uses to implicitly create the mapping table. The name must a
character string with maximum length of 8 bytes.

**blank**

An implicitly defined database is used.

When processing a REORG TABLESPACE SHRLEVEL CHANGE request, the
REORG utility has the option to create its own mapping table and mapping index,
instead of relying on user's input. Specifying this subsystem parameter with a
valid database name directs REORG to allocate the mapping table in the database
that is specified. By default, REORG uses an implicitly defined database for the
mapping table allocation.

**REORG DROP PBG PARTS field (REORG_DROP_PBG_PARTS subsystem parameter)**

The REORG_DROP_PBG_PARTS subsystem parameter specifies whether the
REORG utility removes trailing empty partitions when operating on an entire
partition-by-growth table space. An empty trailing partition occurs when the
REORG utility moves all data records from a partition into lower numbered
partitions.

Acceptable values: DISABLE, ENABLE
Default: DISABLE
Update: option 37 36 on panel DSNTIPB
DSNZPxxx: DSN6SPRM REORG_DROP_PBG_PARTS

This parameter is meaningful only when the REORG utility is run against an entire
partition-by-growth table space. It is ignored for:
- LOB table spaces
- Table spaces that are not defined as partition-by-growth
- Partition-by-growth table spaces that are defined with a MAXPARTITITONS value
  of 1
- Partition-by-growth table spaces that contain tables that are defined with DATA
  CAPTURE CHANGES
- Hash partition-by-growth table spaces

**ENABLE**

Specifies that any trailing empty partitions that are present at the successful
completion of the REORG are always removed. LOB table spaces and auxiliary
indexes that are associated with these empty partition-by-growth partitions are
also removed.

**DISABLE**

Specifies that the number of partition-by-growth partitions at the successful
completion of the REORG are always equal or greater than the number of
partitions before the REORG utility was run. Even if the REORG is able to
relocate all data records into the lowest numbered partitions, trailing empty
partition-by-growth partitions are retained.

**REORG TS NOPAD DEFAULT (REORG_TS_NOPAD_DEFAULT subsystem parameter)**

Specifies the default setting for the NOPAD option of the UNLOAD EXTERNAL
specification or the DISCARD specification of the REORG TABLESPACE utility. In
other words, the setting to use for a REORG TABLESPACE statement that does not
explicitly specify a NOPAD option.
Acceptable values: YES, NO
Default: YES
Update: option 37 on panel DSNTIPB
DSNZPxxx: DSN6PRM REORG_TS_NOPAD_DEFAULT

When the REORG TABLESPACE utility is run with DISCARD or UNLOAD EXTERNAL, the NOPAD option specifies whether variable-length columns in unloaded or discarded records occupy the actual data length without additional padding. The unloaded records can have varying lengths. The REORG_TS_NOPAD_DEFAULT subsystem parameter indicates whether the NOPAD option is NO or YES by default when it is omitted from the utility control statement. Valid options for the REORG_TS_NOPAD_DEFAULT subsystem parameter are:

YES  Specifies that, by default, the variable-length columns in the unloaded or discarded records occupy the actual data length without additional padding.

NO   Specifies that, by default, REORG processing pads variable-length columns in the unloaded or discarded records to their maximum length; the unloaded or discarded records have equal lengths for each table.

Related reference:

 Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)

**DSNTIPP: Protection panel**

The entries on the DSNTIPP panel are related to security matters.

ARCHIVE LOG RACF field (PROTECT subsystem parameter)

The PROTECT subsystem parameter specifies whether archive log data sets are to be protected with individual profiles with RACF when they are created.

Acceptable values: YES, NO
Default: NO
Update: see "DSNTIPP: Protection panel" not during migration
DSNZPxxx: DSN6ARVP PROTECT
If you specify YES, RACF protection must be active for Db2. However, a value of
YES also means that you cannot use RACF generic profiles for archive log data
sets. In addition, RACF class TAPEVOL must be active if your archive log is on
tape. Otherwise, the offload fails.

Related concepts:
- Managing access through RACF (Managing Security)

USE PROTECTION field (AUTH subsystem parameter)
The AUTH subsystem parameter controls whether Db2 is to check authorizations.

Acceptable values: YES, NO
Default: YES
Update: option 38 on panel DSNTIPB
DSNZPxxx: DSN6SPRM AUTH

YES

- Enables authorization checking in Db2.
- If you specified YES for the value of CACHE DYNAMIC SQL on panel
  DSNTIP8, you must also specify YES for this value.

  Recommendation: Specify YES.

NO

- Disables all authorization checking in Db2 and disables the GRANT statement.
- In this case, every privilege is granted to PUBLIC.

Note: This is a security-related parameter. If it is set to NO, individual grants
cannot be performed and all privileges are granted to PUBLIC, which allows any
user to access user data.

PLAN AUTH CACHE field (AUTHCACH subsystem parameter)
The AUTHCACH subsystem parameter specifies the size (in bytes per plan) of the
authorization cache that is to be used if no CACHESIZE is specified on the BIND
PLAN subcommand.

Acceptable values: 0 - 4096 in multiples of 256
Default: 3072
Update: option 38 on panel DSNTIPB
DSNZPxxx: DSN6SPRM AUTHCACH

0

- Specifies that no authorization cache is to be used.

256 - 4096 (integers in multiples of 256)

- For an authorization cache, you need 32 bytes of overhead plus (8 bytes of
  storage times the number of concurrent users).

Note: This is a security-related parameter.

Related tasks:
- Caching authorization IDs for plans (Managing Security)

Related reference:
- BIND PLAN (DSN) (Db2 Commands)
PACKAGE AUTH CACHE field (CACHEPAC subsystem parameter)
The CACHEPAC subsystem parameter determines the amount of storage in bytes to allocate for caching package authorization information for all packages on this Db2 member. The cache is stored in the DSN1DBM1 address space.

Acceptable values: 0 - 10M
Default: 5M
Update: option 38 on panel DSNTIPB
DSNZPxxx: DSN6SPRM CACHEPAC
Security parameter: Yes

Note: This is a security-related parameter. When RACF authorization is used and this parameter is set, the changes that are made in RACF profiles are not immediately reflected. This setting might result in a user (cached AUTHID) being able to run a package.

ROUTINE AUTH CACHE field (CACHERAC subsystem parameter)
The CACHERAC subsystem parameter determines the amount of storage in bytes to allocate for caching routine authorization information for all routines on this Db2 member. Routines include stored procedures and user-defined functions.

Acceptable values: 0 - 10M
Default: 5M
Update: option 38 on panel DSNTIPB
DSNZPxxx: DSN6SPRM CACHERAC
Security parameter: Yes

Note: This is a security-related parameter. When RACF authorization is used and this parameter is set, the changes that are made in RACF profiles are not immediately reflected. This setting might result in a user (cached AUTHID) being able to invoke a routine or function.

AUTH EXIT LIMIT field (AEXITLIM subsystem parameter)
The AEXITLIM subsystem parameter controls the number of abends of the Db2 access control authorization exit routine that are to be tolerated before it is shut down.

Acceptable values: 0 - 32767
Default: 10
Update: option 38 on panel DSNTIPB
DSNZPxxx: DSN6SPRM AEXITLIM

After the exit routine shuts down, it can be reactivated only by restarting Db2. A very low setting for the AUTH EXIT LIMIT field might cause the exit routine to shut down in response to routine abends such as timeouts. A very high setting, on the other hand, may mask a problem with the exit routine environment that can result in degraded Db2 or system performance.

AUTH EXIT CHECK (AUTHEXIT_CHECK subsystem parameter)
The AUTHEXIT_CHECK subsystem parameter specifies whether the owner or the primary authorization ID is used for authorization checks when the access control authorization exit (DSNX@XAC) is active.
Acceptable values: PRIMARY, DB2
Default: PRIMARY
Update: No
DSNZPxxx: DSN6SPRM.AUTHEXIT_CHECK
Data sharing scope: Group
Security parameter: Yes

**PRIMARY**

Specifies that Db2 provides the ACEE of the primary authorization ID to perform all authorization checks. The primary authorization ID must be permitted access to the resources in RACF. This is the default value for the field.

**DB2**

Specifies that Db2 provides the ACEE of the package or plan owner to perform authorization checking when processing the autobind, BIND and REBIND commands and, if needed, during the execution of the package or plan. Db2 also provides the ACEE of the authorization ID as determined by the DYNAMICRULES option to perform dynamic SQL authorization checking at run time. The access control authorization exit uses the ACEE for XAPLUCHK for authorization checking. The XAPLUCHK authorization ID can be a user or a group in RACF.

To ensure successful authorization checks with the owner ACEE, the owner authorization ID in XAPLUCHK must be permitted access to the resources in RACF. If the owner is a group in RACF, you need to permit the group access to the resource associated with the connection in the RACF DSNR class. You can issue the PERMIT command to grant a group access to `subsystem.BATCH` in the DSNR class, as follows:

```
PERMIT DSN.BATCH CLASS(DSNR) ID(DB2GRP) ACCESS(READ)
```

**AUTH EXIT CACHE REFR (AUTHEXIT_CACHEREFRESH subsystem parameter)**

The AUTHEXIT_CACHEREFRESH subsystem parameter specifies whether the cache entries of the package authorization, the routine authorization, the DDF user authentication, and the dynamic statement are refreshed and whether the dependent packages are invalidated when a user profile or resource access is changed in RACF. The cache entries are refreshed only when the access control authorization exit (DSNX@XAC) is active.

Acceptable values: ALL, NONE
Default: NONE
Update: No
DSNZPxxx: DSN6SPRM.AUTHEXIT_CACHEREFRESH
Data sharing scope: Group
Security parameter: Yes

**ALL**

Specifies that Db2 refreshes the cache entries of the package authorization, the routine authorization, and the dynamic statement and invalidates dependent packages when the user profile or resource access is changed in RACF. The cache entries are refreshed only when the access control authorization exit is active.

**NONE**

Specifies that Db2 does not refresh the cache entries of the package authorization, the routine authorization, and the dynamic statement or
invalidate dependent packages when the user profile or resource access is changed in RACF. This is the default value for the field.

When the AUTHEXIT_CACHEREFRESH subsystem parameter is set to ALL and the access control authorization exit is active, Db2 listens to type 62, type 71, and type 79 ENF signals from RACF for user profile or resource access changes and refreshes the Db2 cache entries accordingly. If you define RACF classes for Db2 objects and administrative authorities without using IBM-supplied RACF resource classes, you need to enable the SIGNAL=YES option for these classes in the RACF Class Descriptor Table.

Related reference:
- [Invalid and inoperative packages (Managing Security)](#)
- [Db2 processing of ENF signals (Managing Security)](#)

**DSNTIPP1: Protection panel 2**

Use the DSNTIPP1 panel to specify security options, including the authorization IDs that are required for system administrators and system operators, and the authorization IDs or roles that are required for security administrators. You can also specify the authorization IDs that are to be used to install, configure, and validate Db2-supplied routines.

Because space on this panel is limited, some input fields are not wide enough to display the entire contents of the field at one time. These fields are defined as ISPF scrollable fields, which let you see the field contents. A less-than symbol (<) or a greater-than (>) symbol denotes a scrollable field. A > symbol indicates that the field can be scrolled to the right, and a < symbol indicates that the field can be scrolled to the left. You can position your cursor in the field and use ISPF scroll commands LEFT and RIGHT to scroll through the field. To see the entire contents of the field at one time, type EXPAND in the command field, position your cursor in the scrollable field, and press Enter. (Alternatively, you can assign your PF keys to be the LEFT, RIGHT, and EXPAND commands.)

---

**Figure 44. Protection panel 2: DSNTIPP1**

---
**SYSTEM ADMIN 1 field (SYSADM subsystem parameter)**
The SYSADM subsystem parameter specifies the first of two authorization IDs that are to have installation SYSADM authority.

Acceptable values: 1 - 8 characters, the first of which must be an alphabetic character
Default: SYSADM
Update: option 39 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SYSADM
Security parameter: Yes

The two users with installation SYSADM authority are permitted access to Db2 in all cases.

Note: This is a security-related parameter. The authorization ID that is specified for this parameter can manage the Db2 subsystem even when the Db2 catalog is unavailable. This authorization ID can access all user data and can run any application.

**Related reference:**

[Installation SYSADM (Managing Security)]

**SYSTEM ADMIN 2 field (SYSADM2 subsystem parameter)**
The SYSADM2 subsystem parameter specifies the second of two authorization IDs that are to have installation SYSADM authority.

Acceptable values: 1 - 8 characters, the first of which must be an alphabetic character
Default: SYSADM
Update: option 39 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SYSADM2
Security parameter: Yes

If you leave this field blank, the value is set to the value of the SYSTEM ADMIN 1 field.

Note: This is a security-related parameter. The authorization ID that is specified for this parameter can manage the Db2 subsystem even when the Db2 catalog is unavailable. This authorization ID can access all user data and can run any application.

**SYSTEM OPERATOR 1 field (SYSOPR1 subsystem parameter)**
The SYSOPR1 subsystem parameter specifies the first of two authorization IDs that are to have installation SYSOPR authority.

Acceptable values: 1 - 8 characters, the first of which must be an alphabetic character
Default: SYSOPR
Update: option 39 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SYSOPR1
Security parameter: Yes

The two users with installation SYSOPR authority are permitted access to Db2 even if the Db2 catalog is unavailable.
The value of the SYSTEM OPERATOR 1 field or the SYSTEM OPERATOR 2 field is used for authorization during installation or migration if you install or migrate your Db2 subsystem without SYSADM authority.

If you leave this field blank, the value is set to the value of the SYSTEM ADMIN 1 field.

**Recommendation:** Set the value of this field or the value of the SYSTEM OPERATOR 2 field to SYSOPR. Doing so ensures that Db2 commands that are issued from the console can be processed correctly when the Db2 catalog is unavailable.

**Note:** This is a security-related parameter. The authorization ID that is specified for this parameter can issue most of the Db2 commands such as BIND QUERY, CANCEL THREAD, and STOP DB2. This ability might provide the authorization ID with information about user data. Also, the authorization ID can run the DSN1SDMP utility and terminate any utility job.

**Related reference:**

[Installation SYSOPR (Managing Security)]

**SYSTEM OPERATOR 2 field (SYSOPR2 subsystem parameter)**

The SYSOPR2 subsystem parameter specifies the second of two system operators that are to have installation SYSOPR authority.

**Acceptable values:** 1 - 8 characters, the first of which must be an alphabetic character

**Default:** SYSOPR

**Update:** option 39 on panel DSNTPB

**DSNZPxxx:** DSN6SPRM SYSOPR2

**Security parameter:** Yes

The value of the SYSTEM OPERATOR 1 field or the SYSTEM OPERATOR 2 field is used for authorization during installation or migration if you install or migrate your Db2 subsystem without SYSADM authority.

If you leave this field blank, the value is set to the value of the SYSTEM OPERATOR 1 field.

**Recommendation:** Set the value of this field or the value of the SYSTEM OPERATOR 1 field to SYSOPR. Doing so ensures that Db2 commands that are issued from the console can be processed correctly when the Db2 catalog is unavailable.

**Note:** This is a security-related parameter. The authorization ID that is specified for this parameter can issue most of the Db2 commands such as BIND QUERY, CANCEL THREAD, and STOP DB2. This ability might provide the authorization ID with information about user data. Also, the authorization ID can run the DSN1SDMP utility and terminate any utility job.

**SECURITY ADMIN 1 field (SECDAM1 subsystem parameter)**

The SECDAM1 subsystem parameter specifies the first of two authorization IDs or roles that are to have Db2 security administrator authority. In the SEC ADMIN 1 TYPE field, specify whether this entry is an authorization ID or a role.
If you want to separate Db2 security administrator duties from system administrator duties for this subsystem, set at least one SECADM subsystem parameter to an authorization ID, or create the necessary trusted contexts and roles before setting the SEPARATE SECURITY field to YES. If you specify YES for SEPARATE SECURITY, system administrator authority can no longer be used to perform security tasks, and the SECADM authority is required to manage security objects such as trusted contexts and roles. If both SECADM subsystem parameters are set to roles and those roles have not been created, no one will have the authority to manage security objects.

If the access control authorization exit routine (DSNX@XAC) is active, then the exit routine is called to check for SECADM authorization and this subsystem parameter is not checked.

**Note:** This is a security-related parameter. A user that has SECADM authority can manage security-related objects such as trusted contexts, roles, and column masks. The user can also grant privileges and revoke privileges that are granted by others.

### Related concepts:
- Access control authorization exit routine (Managing Security)

### Related tasks:
- Installation step 10: Establish subsystem security (optional)

### Related reference:
- SEPARATE SECURITY field (SEPARATE_SECURITY subsystem parameter)
- SEC ADMIN 1 TYPE field (SECADM1_TYPE subsystem parameter)
- SECADM1_INPUT_STYLE in macro DSN6SPRM

### Related information:
- Authorization IDs and roles (Managing Security)

### SEC ADMIN 1 TYPE field (SECADM1_TYPE subsystem parameter)

The SECADM1_TYPE subsystem parameter specifies whether the entry in the SECURITY ADMIN 1 field is an authorization ID or a role.
**AUTHID**

The SECADM1 value is an authorization ID.

**ROLE**

The SECADM1 value is a role.

**Note:** This is a security-related parameter. The value specifies whether the SECADM1 parameter is an authorization ID or a role. Changing the value of this field changes the SECADM1 parameter from an authorization ID to a role, or from a role to an authorization ID.

**Related reference:**

- SECURITY ADMIN 1 field (SECADM1 subsystem parameter)
- SECADM1_INPUT_STYLE in macro DSN6SPRM

**SECURITY ADMIN 2 field (SECADM2 subsystem parameter)**

The SECADM2 subsystem parameter specifies the second of two authorization IDs or roles that are to have Db2 security administrator authority. In the SEC ADMIN 2 TYPE field, specify whether this entry is an authorization ID or a role.

**Acceptable values:**

If SEC ADMIN 2 TYPE is AUTHID: 1 - 8 characters, starting with an alphabetic character.

If SEC ADMIN 2 TYPE is ROLE: an ordinary SQL identifier (up to 128 characters) that designates a role. The role identifier cannot begin with 'SYS' and cannot be ACCESSCTRL, DATAACCESS, DBADM, DBCTRL, DBMAINT, NONE, NULL, PACKADM, PUBLIC, SECADM, or SQLADM.

**Default:** SECADM

**Update:** option 39 on panel DSNTIPB

**DSNZPxxx:** DSN6SPRM SECADM2

**Security parameter:** Yes

If you leave this field blank, the value is set to the value of the SECURITY ADMIN 1 field.

If you want to separate Db2 security administrator duties from system administrator duties for this subsystem, set at least one SECADM subsystem parameter to an authorization ID, or create the necessary trusted contexts and roles before setting the SEPARATE SECURITY field to YES. If you specify YES for SEPARATE SECURITY, system administrator authority can no longer be used to perform security tasks, and the SECADM authority is required to manage security objects such as trusted contexts and roles. If both SECADM subsystem parameters are set to roles and those roles have not been created, no one will have the authority to manage security objects.

If the access control authorization exit (DSNX@XAC) is active, then the exit is called to check for SECADM authorization and this parameter is not checked.
Note: This is a security-related parameter. A user that has SECADM authority can manage security-related objects such as trusted contexts, roles, and column masks. The user can also grant privileges and revoke privileges that are granted by others.

Related concepts:
- Access control authorization exit routine (Managing Security)

Related tasks:
- Installation step 10: Establish subsystem security (optional)

Related information:
- Authorization IDs and roles (Managing Security)

SEC ADMIN 2 TYPE field (SEACDM2_TYPE subsystem parameter)
The SEACDM2_TYPE subsystem parameter specifies whether the entry in the SECURITY ADMIN 2 field is an authorization ID or a role.

Acceptable values: AUTHID, ROLE
Default: AUTHID
Update: option 39 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SEACDM2_TYPE
Security parameter: Yes

AUTHID
The SEACDM2 value is an authorization ID.

ROLE
The SEACDM2 value is a role.

Note: This is a security-related parameter. The value specifies whether the SEACDM2 parameter is an authorization ID or a role. Changing the value of this field changes the SEACDM2 parameter from an authorization ID to a role, or from a role to an authorization ID.

Related reference:
- SECURITY ADMIN 2 field (SEACDM2 subsystem parameter)
- SEACDM2_INPUT_STYLE in macro DSN6SPRM

Related information:
- Authorization IDs and roles (Managing Security)
- Managing access through trusted contexts (Managing Security)

SEPARATE SECURITY field (SEPARATE_SECURITY subsystem parameter)
The SEPARATE_SECURITY subsystem parameter specifies whether Db2 security administrator duties are to be separated from system administrator duties for this subsystem.

Acceptable values: NO, YES
Default: NO
Update: option 39 on panel DSNTIPB
DSNZPxxx: DSN6SPRM SEPARATE_SECURITY
Security parameter: Yes

NO Overlaps Db2 security administrator duties and system administrator duties. If you are migrating and want SYSADM and SYSCTRL authorities to remain
unchanged, use this setting. Users with SYSADM authority can manage all
security objects, perform grants, and revoke privileges that are granted by
others. Users with SYSCTRL authority can manage roles, perform most grants,
and revoke privileges that are granted by others.

YES
Separates Db2 security administrator duties from system administrator duties.
Users with SYSADM authority cannot manage security objects (such as roles
and trusted contexts), perform grants, or revoke privileges granted by others.
Users with SYSCTRL authority cannot manage roles, perform grants, or revoke
privileges granted by others. However, existing grants made by users with
SYSADM or SYSCTRL authority are unchanged. SEADM or ACCESSCTRL
authority is required for security administration. The SEPARATE_SECURITY
subsystem parameter does not apply to or affect users with installation
SYSADM authority.

Before setting SEPARATE_SECURITY to YES, set at least one SEADM system
parameter to an authorization ID, or create the necessary trusted contexts and
roles. If you specify YES, system administrator authority can no longer be used
to perform security tasks, and the SEADM authority is required to manage
security objects such as trusted contexts and roles. If both SEADM system
parameters are set to roles and those roles have not been created, no one will
have the authority to manage security objects.

Note: This is a security-related parameter.

Related tasks:
[Installation step 10: Establish subsystem security (optional)]

UNKNOWN AUTHID field (DEFLTID subsystem parameter)
The DEFLTID subsystem parameter specifies the authorization ID that is to be
used if RACF is not available for batch access and USER= is not specified in the
JOB statement.

Acceptable values: 1 - 8 characters, the first of which must be an alphabetic
coloracter
Default: IBMUSER
Update: option 39 on panel DSNTPB
DSNZPxxx: DSN6SPRM DEFLTID

RESOURCE AUTHID field (RLAUTH subsystem parameter)
The RLAUTH subsystem parameter specifies the authorization ID that is to be
used if you plan to use the resource limit facility (governor).

Acceptable values: 1 - 8 characters
Default: SYSIBM
Update: option 39 on panel DSNTPB
DSNZPxxx: DSN6SYSP RLAUTH

BIND NEW PACKAGE field (BINDNV subsystem parameter)
The BINDNV subsystem parameter controls whether BIND or BINDADD authority
is to be required for a user to bind a new version of an existing package.

Acceptable values: BINDADD, BIND
Default: BINDADD
Update: option 39 on panel DSNTPB
DSNZPxxx: DSN6SPRM BINDNV
Security parameter: Yes

If you accept the default, BINDADD, only users with the BINDADD system privilege are allowed to create a new package. If you specify BIND, users with the BIND privilege on a package or collection are allowed to create a new version of an existing package when they bind it. Users with PACKADM authority are also allowed to add a new package or a new version of a package to a collection.

Note: This is a security-related parameter. When this parameter is set to BIND, the BINDADD privilege check is bypassed, and a user with PACKADM authority can create new packages.

Related reference:

Privileges required for handling plans and packages (Managing Security)

DBADM CREATE AUTH field (DBACRVW subsystem parameter)
The DBACRVW subsystem parameter controls whether an authorization ID with DBADM authority on a database is to be allowed to complete certain tasks.

Acceptable values: NO, YES
Default: NO
Update: option 39 on panel DSNTIPB
DSNZPxxx: DSN6SPRM DBACRVW
Security parameter: Yes

NO Does not allow authorization IDs with DBADM authority on a database to complete the following tasks:

- Create a view for another authorization ID on tables in that database.
- Create a materialized query table or alter a table to become a materialized query table for another authorization ID.
- Create an alias for itself or another authorization ID for a table in that database.

YES Allows authorization IDs with DBADM authority on a database to complete the following tasks:

- Create a view for another authorization ID on tables in that database.
- Create a materialized query table or alter a table to become a materialized query table for another authorization ID. This action requires that DBADM authority is held on the database in which the tables of the fullselect reside and that the authorization ID has DBADM authority on the database in which the materialized query table is to reside.
- Create an alias for itself or another authorization ID for a table in that database.

If you specify YES, an authorization ID with DBCTRL authority on a database can also create an alias for itself or for another authorization ID for a table in any database.

Specifying YES results in less need for SYSADM authority on a database. However, users that need full authority may still need to have SYSADM authority. Specifying YES does not allow an authorization ID with DBADM authority to grant authority on that view.
**Note:** This is a security-related parameter. It controls whether an authorization ID or role with DBADM authority on a database is to be allowed to create views, aliases, and materialized query tables for another authorization ID or role. If it is set to YES, authorization IDs and roles with DBADM authority on a database can give others access to user data.

**REVOKE DEP PRIV field (REVOKE_DEP_PRIVILEGES subsystem parameter)**

The REVOKE_DEP_PRIVILEGES subsystem parameter controls whether revoking a privilege from a user is to cause dependent privileges to be revoked. If dependent privileges are to be revoked, revoking a privilege from a user also revokes the privilege from anyone that the user has granted that privilege to.

- **Acceptable values:** NO, YES, SQLSTMT
- **Default:** SQLSTMT
- **Update:** option 39 on panel DSNTIP8
- **DSNZPxxx:** DSN6SPRM REVOKE_DEP_PRIVILEGES
- **Security parameter:** Y

**NO** REVOKE statements do not include dependent privileges. An error occurs if a REVOKE statement contains the INCLUDING DEPENDENT PRIVILEGES clause.

**YES**

REVOKE statements include dependent privileges, except when ACCESSCTRL, DATAACCESS, and system DBADM authorities are revoked. An error occurs if a REVOKE statement contains the NOT INCLUDING DEPENDENT PRIVILEGES clause, except when ACCESSCTRL, DATAACCESS, and system DBADM authorities are revoked.

**SQLSTMT**

Allows revoking of dependent privileges to be controlled at the SQL level, as specified in REVOKE statements. Db2 recognizes the dependent privileges clause (INCLUDING DEPENDENT PRIVILEGES or NOT INCLUDING DEPENDENT PRIVILEGES) of the REVOKE statement.

**Note:** This is a security-related parameter. If it is set to NO, privileges that were granted by a user are retained even if that user loses the authority that allowed the user to perform the grant.

**DSNTIPM: MVS PARMLIB updates panel**

The entries on the DSNTIPM panel produce the DSNTIJMV job, which defines Db2 to z/OS and updates several PARMLIB members.

DSNTIJMV updates the following PARMLIB members:
- IEFSSNxx, to define Db2 and IRLM as formal z/OS subsystems
- IEAAPFx, to authorize the prefix.SDSNLOAD, prefix.SDSNLINK, and prefix.SDSNEXIT libraries
- LNKLISTxx, to include the prefix.SDSNLINK library.

**Updating the parameters:** Different sites have different requirements for identifying Db2 to z/OS; as a result, the updates that DSNTIJMV makes to z/OS PARMLIB members might be incomplete. To ensure that the updates are complete, it is recommended that you edit the z/OS PARMLIB members directly when you install or migrate Db2. This is substantially easier than editing DSNTIJMV.
The SUBSYSTEM NAME field specifies the z/OS subsystem name for Db2.

Acceptable values: 1 - 4 characters, the first of which must be a letter (A - Z), #, $, or @. Others must be letters A - Z, integers 0 - 9, #, $, or @.

Default: DSN1

Update: see below
dsnhdecp: SSID

The name is used in member IEFSSNxx of SYS1.PARMLIB.

If you specified a group attachment name in the GROUP ATTACH field on panel DSNTIPK, the SSID DECP value is set to that value.

**Updating the parameters:** Different sites have different requirements for identifying Db2 to z/OS; as a result, the updates that DSNTIJMV makes to z/OS PARMLIB members might be incomplete. To ensure that the updates are complete, you should edit the z/OS PARMLIB members directly when you install or migrate Db2. This is substantially easier than editing DSNTIJMV.

**COMMAND PREFIX field**

The value of the COMMAND PREFIX field specifies the Db2 subsystem command prefix. When the prefix appears at the beginning of a command that is entered at a z/OS operator's console, z/OS passes the command to Db2 for processing. The command prefix is used in the Db2 entry of member IEFSSNxx of SYS1.PARMLIB.

Acceptable values: 1 - 8 characters; the first character must be a non-alphanumeric character.

Default: -DSN1 (hyphen, concatenated with subsystem name)

Update: see "DSNTIPM: MVS PARMLIB updates panel" on page 273

DSNZPxxx: none
The first character of the command prefix must be a character from the table below. The remaining characters of the command prefix must be from the table below, letters A - Z, or numbers 0 - 9.

### Table 32. Allowable special characters for the command prefix

<table>
<thead>
<tr>
<th>Name</th>
<th>Character</th>
<th>Hexadecimal representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>cent sign</td>
<td>¢</td>
<td>X'4A'</td>
</tr>
<tr>
<td>period</td>
<td>.</td>
<td>X'4B'</td>
</tr>
<tr>
<td>less-than sign</td>
<td>&lt;</td>
<td>X'4C'</td>
</tr>
<tr>
<td>plus sign</td>
<td>+</td>
<td>X'4E'</td>
</tr>
<tr>
<td>vertical bar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ampersand</td>
<td>&amp;</td>
<td>X'50'</td>
</tr>
<tr>
<td>exclamation point</td>
<td>!</td>
<td>X'5A'</td>
</tr>
<tr>
<td>dollar sign</td>
<td>$</td>
<td>X'5B'</td>
</tr>
<tr>
<td>asterisk</td>
<td>*</td>
<td>X'5C'</td>
</tr>
<tr>
<td>right parenthesis</td>
<td>)</td>
<td>X'5D'</td>
</tr>
<tr>
<td>semi-colon</td>
<td>;</td>
<td>X'5E'</td>
</tr>
<tr>
<td>hyphen</td>
<td>-</td>
<td>X'60'</td>
</tr>
<tr>
<td>slash</td>
<td>/</td>
<td>X'61'</td>
</tr>
<tr>
<td>percent sign</td>
<td>%</td>
<td>X'6C'</td>
</tr>
<tr>
<td>underscore</td>
<td>_</td>
<td>X'6D'</td>
</tr>
<tr>
<td>question mark</td>
<td>?</td>
<td>X'6F'</td>
</tr>
<tr>
<td>colon</td>
<td>;</td>
<td>X'7A'</td>
</tr>
<tr>
<td>number sign</td>
<td>#</td>
<td>X'7B'</td>
</tr>
<tr>
<td>at sign</td>
<td>@</td>
<td>X'7C'</td>
</tr>
<tr>
<td>apostrophe</td>
<td>'</td>
<td>X'7D'</td>
</tr>
<tr>
<td>equal sign</td>
<td>=</td>
<td>X'7E'</td>
</tr>
<tr>
<td>quotation marks</td>
<td>&quot;</td>
<td>X'7F'</td>
</tr>
</tbody>
</table>

1. To use the ampersand (&), accept the default in this field, and then edit job DSNTIJMV to specify the ampersand as the command prefix.
2. To use the apostrophe (‘), you must code two consecutive apostrophes in your IEFSSNxx member. For example, the entry for subsystem DB2A with a command prefix of 'DB2A and a scope of started looks like this:

   `DB2A,DSN3IN1,'DSN3EPX,' 'DB2A,'`

Do not use the JES2 backspace character as a command prefix character. Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or z/OS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix beginning from the first character. For example, you cannot assign - to one subsystem and -DB2A to another. Similarly, you cannot assign ?DB2 to one subsystem and ?DB2A to another. However, you can assign -DB2A and -DB2B to different Db2 subsystems.

To use multiple-character command prefixes, have the system programmer update the IEFSSNxx subsystem definition statements in SYS1.PARMLIB.

### Related concepts:
Advantages of database access threads in INACTIVE mode (Db2 Performance)

SUBSYSTEM MEMBER field
The value of the SUBSYSTEM MEMBER field specifies the last two characters (xx) of the name of member IEFSSNxx of SYS1.PARMLIB.

Acceptable values: 2 alphanumeric characters
Default: 00
Update: see DSNTIPM: MVS PARMLIB updates panel
DSNZPxxx: none

The subsystem member name indicates the available z/OS subsystems, including Db2 and IRLM.

SUBSYSTEM SEQUENCE field
The SUBSYSTEM SEQUENCE field specifies a sequence number that is to be used for insertion. Use any number that is greater than the highest sequence number that is already used in the IEFSSNxx PARMLIB member.

Acceptable values: 1 - 99999993
Default: 88888888
Update: see DSNTIPM: MVS PARMLIB updates panel
DSNZPxxx: none

AUTH MEMBER field
The value of the AUTH MEMBER field specifies the last two characters (xx) of the name of member IEAAPFx of SYS1.PARMLIB. This member is to be used for authorized program facility (APF) authorization of the prefix.SDSNLOAD, prefix.SDSNLINK, and prefix.SDSNEXIT libraries.

Acceptable values: 2 alphanumeric characters
Default: 00
Update: see DSNTIPM: MVS PARMLIB updates panel
DSNZPxxx: none

The specified data set must be APF-authorized. The member name must currently exist for the z/OS update job DSNTIJMV to work correctly.

You can use the PROGxx member instead of the IEAAPFx member. In this case, you must manually name the PROGxx member because job DSNTIJMV does not do it for you.

AUTH SEQUENCE field
The value of the AUTH SEQUENCE field is to be used as the sequence number for insertion in the IEAAPFx PARMLIB member. Specify any number that is greater than the highest sequence number that is already used in the IEAAPFx PARMLIB member.

Acceptable values: 1 - 99999995
Default: 88888888
Update: see DSNTIPM: MVS PARMLIB updates panel
DSNZPxxx: none
**LINK LIST ENTRY field**

The value of the LINK LIST ENTRY field specifies the last two characters of LNKLSTxx as needed to include the prefix.SDNSLINK library.

Acceptable values: 2 alphanumeric characters
Default: 00
Update: see “DSNTIPM: MVS PARMLIB updates panel” on page 273
DSNZPxxx: none

**LINK LIST SEQUENCE field**

The LINK LIST SEQUENCE field specifies the sequence number that is to be used for insertion. Use any number that is greater than the highest sequence number that is already used in the LNKLSTxx PARMLIB member.

Acceptable values: 1 - 99999999
Default: 88888888
Update: see [DSNTIPM: MVS PARMLIB updates panel](#)
DSNZPxxx: none

**COMMAND SCOPE field**

The value of the COMMAND SCOPE field controls whether the scope of the command prefix is to be one z/OS system or the entire Sysplex.

Acceptable values: SYSTEM, SYSPLEX, STARTED
Default: STARTED
Update: see [DSNTIPM: MVS PARMLIB updates panel](#)
DSNZPxxx: none

- **SYSTEM**
  Sets the scope of commands to one z/OS system. The command prefix is registered at z/OS IPL.

- **SYSPLEX**
  Sets the scope of commands to the entire Sysplex. The command prefix is registered at z/OS IPL.

- **STARTED**
  Sets the scope of commands to the entire Sysplex. The command prefix is registered at Db2 startup and unregistered when Db2 stops.

  Although STARTED specifies a Sysplex scope, you can also use it for a Db2 subsystem in a non-data-sharing environment. Use STARTED if you intend to use the z/OS automatic restart manager, or if you might move this Db2 into a data sharing group.

**SUPPRESS SOFT ERRORS field (SUPERRS subsystem parameter)**

The SUPERRS subsystem parameter specifies whether Db2 is to record errors such as invalid decimal data and arithmetic exceptions and issue SQLCODEs for them. This option enables or disables the recording of these errors in the operating system data set, SYS1.LOGREC.

Acceptable values: YES, NO
Default: YES
Update: option 40 on panel DSNTIPB
DSNZPxxx: SUPERRS
DSNTIPL: Active log data set parameters

The entries on the DSNTIPL panel define characteristics of active log data sets.

Performance note: Several fields on this panel affect the Db2 use of logging. Be careful when you determine the values that are associated with fields on this panel. These values can greatly affect the performance of your Db2 subsystem.

Related concepts:

How the initial Db2 logging environment is established (Db2 Administration Guide)

NUMBER OF LOGS field

The NUMBER OF LOGS field specifies the number of data sets that are to be established for each copy of the active log during installation.

Acceptable values: 2 - 93
Default: 3
Update: cannot change during update or migration

You can later use the DSNJU003 utility to change the number of log data sets. If you use the DSNJU003 utility to modify the number of logs, your modified value is not reflected on this panel.

Related tasks:

Updating subsystem parameters that cannot be updated by using DSNTIPB

OUTPUT BUFFER field (OUTBUFF subsystem parameter)

The OUTPUT BUFFER field sets the OUTBUFF subsystem parameter, which specifies the size of the output buffer for writing active log data sets.

Acceptable values: OUTPUT BUFFER field: 409600 - 409600000, 400K - 4000000K, 1M- 390M
Default: OUTPUT BUFFER subsystem parameter: 400 - 400000
Update: option 41 on panel DSNTIPB

DSNZPxxx: DSN6LOGP OUTBUFF

Figure 46. Log data sets panel: DSNTIPL
The size in bytes of the output buffer for writing active log data sets, when specified in the OUTPUT BUFFER field.

The size in kilobytes of the output buffer for writing active log data sets, when specified in the OUTPUT BUFFER field.

The size in megabytes of the output buffer for writing active log data sets, when specified in the OUTPUT BUFFER field.

The size in kilobytes of the output buffer for writing active log data sets, when specified for the OUTBUFF subsystem parameter. Omit “K” in this context.

The larger the output buffer, the more likely that a log read request can return the required log records without needing I/O operations. This is a particularly important consideration if the Db2 subsystem is to be used with a data replication product that uses IFCID 306. In Db2 12, log buffers are permanently page-fixed. When you estimate real storage usage, you must use the entire size that you specify for the OUTBUFF parameter. To avoid page-fixing more storage than necessary, carefully choose the setting for OUTBUFF.

**ARCHIVE LOG FREQ field**

The value of the ARCHIVE LOG FREQ field determines, in hours, the interval at which the active log is to be offloaded to the archive log.

| Acceptable values: | 1 - 200 |
| Default:           | 24      |
| Update:            | cannot change during update |
| DSNZPxxx:          | none    |

If you accept the default value of 24, the active log is offloaded approximately once each day.

**UPDATE RATE field**

The value of the UPDATE RATE field specifies the estimated average number of inserts, updates, and deletes that are expected per hour in your subsystem.

| Acceptable values: | 1 - 16,777,216 |
| Default:           | 3600          |
| Update:            | cannot change during update |
| DSNZPxxx:          | none          |

You can use K (as in 32K) for multiples of 1024 bytes and M (as in 16M) for multiples of 1,048,576 bytes.

The size calculations in the DSNTINST CLIST assume that about 400 bytes of data are logged for each insert, update, and delete. The amount of data that is logged for these changes might be different at your site. Therefore, consider changing the size of the log data sets after you gain some experience with Db2 and have a better idea of how many bytes of data are logged for each change. Generally, if you have a subsystem that is tuned for maximum efficiency, you can expect to log about 10 GB of data per hour while processing several millions of updates and inserts.
Together, the UPDATE RATE and the ARCHIVE LOG FREQ (field 3) determine the size of the active logs.

**REMOTE COPY SW ACCEL field (REMOTE_COPY_SW_ACCEL subsystem parameter)**

The REMOTE COPY SW ACCEL field specifies whether Db2 uses software to control the remote copy process for active log output in peer-to-peer remote copy (PPRC) environments.

Acceptable values: DISABLE, ENABLE

Default: DISABLE

Data sharing scope: It is recommended but not required that all members use the same setting. DISABLE is recommended if the active log data sets are not on PPRC-enabled devices.

Online changeable: Yes

DSNZPxxx: DSN6LOGP.REMOTE_COPY_SW_ACCEL

DISABLE

The hardware PPRC function is used if available; this setting is appropriate for non-PPRC environments.

ENABLE

Host software is used. This option can improve overall Db2 performance in PPRC environments at the expense of some additional SRB CPU time in the xxxxMSTR address space. z/OS Version 2 Release 1 or later is required to support this function.

Related concepts:

[Storage servers and advanced features (Db2 Performance)]

**DSNTIPL1: Checkpoint parameters**

The entries on the DSNTIPL1 panel define characteristics for log checkpoints.

```
<table>
<thead>
<tr>
<th>DSNTIPL1</th>
<th>INSTALL DB2 - CHECKPOINT PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter log checkpoint options:</td>
</tr>
<tr>
<td>1 CHECKPOINT TYPE</td>
<td>===&gt; MINUTES</td>
</tr>
<tr>
<td>2 RECORDS/CHECKPOINT</td>
<td>===&gt; NOTUSED</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3 MINUTES/CHECKPOINT</td>
<td>===&gt; 5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4 UR CHECK FREQ</td>
<td>===&gt; 5</td>
</tr>
<tr>
<td>5 UR LOG WRITE CHECK</td>
<td>===&gt; 10K</td>
</tr>
<tr>
<td>6 LIMIT BACKOUT</td>
<td>===&gt; AUTO</td>
</tr>
<tr>
<td>7 BACKOUT DURATION</td>
<td>===&gt; 5</td>
</tr>
<tr>
<td>8 RO SWITCH CHKPTS</td>
<td>===&gt; 10</td>
</tr>
<tr>
<td>9 RO SWITCH TIME</td>
<td>===&gt; 10</td>
</tr>
<tr>
<td>10 LEVELID UPDATE FREQ</td>
<td>===&gt; ON</td>
</tr>
</tbody>
</table>
```

PRESS: ENTER to continue  RETURN to exit  HELP for more information
CHECKPOINT TYPE field (CHKTYPE subsystem parameter)

The CHKTYPE subsystem parameter indicates whether the interval between log checkpoints is to be based on the number of written log records, the time between checkpoints, or both.

Acceptable values: LOGRECS, MINUTES, BOTH
Default: MINUTES
Update: option 42 on panel DSNTIPB
DSNZPxxx: DSN6SYSP CHKTYPE

The following values can be specified in the CHECKPOINT TYPE field on panel DSNTIPL1. If you run the installation or migration CLIST, the setting of the CHECKPOINT TYPE field determines the setting of the CHKTYPE subsystem parameter. The setting of the CHECKPOINT TYPE field also defines which settings are allowed in the RECORDS/CHECKPOINT and MINUTES/CHECKPOINT fields.

LOGRECS
Log checkpoints occur after a certain number of log records are created, as specified by the CHKFREQ subsystem parameter. The CHKFREQ subsystem parameter is set through the RECORDS/CHECKPOINT field, and the CHKLOGR and CHKMINS subsystem parameters are both assigned a value of NOTUSED. When LOGRECS is specified in the CHECKPOINT TYPE field, the CHKTYPE subsystem parameter is set to SINGLE.

MINUTES
Log checkpoints occur at an interval of a certain number of minutes, as specified by the CHKFREQ subsystem parameter. The CHKFREQ subsystem parameter is set through the MINUTES/CHECKPOINT field, and the CHKLOGR and CHKMINS subsystem parameters are both assigned a value of NOTUSED. When MINUTES is specified in the CHECKPOINT TYPE field, the CHKTYPE subsystem parameter is set to SINGLE.

BOTH
Log checkpoints occur after a specified number of log records are created or after a specified number of minutes, whichever occurs first. The number of log records is specified by the CHKLOGR subsystem parameter and the number of minutes is specified by the CHKMINS subsystem parameter. These subsystem parameters are set through the RECORDS/CHECKPOINT and MINUTES/CHECKPOINT fields, respectively. The CHKFREQ parameter is assigned a value of NOTUSED. When BOTH is specified in the CHECKPOINT TYPE field, the CHKTYPE subsystem parameter is set to BOTH.

The following settings are valid for the CHKTYPE subsystem parameter:

SINGLE
The Db2 subsystem uses the CHKFREQ subsystem parameter setting to determine one of the following numbers:

- The number of log records between checkpoints, if the setting is an integer from 1000 to 16000000. CHKFREQ settings from 200 - 999 are automatically increased to 1000.
- The number of minutes between checkpoints, if the setting is an integer from 1 - 60.
The CHKLOGR and CHKMINS subsystem parameters must both be set to NOTUSED.

**BOTH** Log checkpoints occur after a specified number of log records are created or after a specified number of minutes, whichever occurs first. The number of log records is specified by the CHKLOGR subsystem parameter and the number of minutes is specified by the CHKMINS subsystem parameter. The CHKFREQ subsystem parameter must be set to NOTUSED.

**RECORDS/CHECKPOINT field (CHKFREQ and CHKLOGR subsystem parameters)**

The RECORDS/CHECKPOINT field determines the approximate number of log records that are to be created between scheduled log checkpoints.

- **Acceptable values:**
  - 1 000 - 16 000 000 (if CHECKPOINT TYPE is LOGRECS);
  - NOTUSED (if CHECKPOINT TYPE is MINUTES);
  - 1 000 - 99 999 999 (if CHECKPOINT TYPE is BOTH)

**Default:** NOTUSED

**Update:** option 42 on panel DSNTIPB

**DSNZP:** DSN6SYS.P.CHKFREQ and DSN6SYS.P.CHKLOGR

- If the CHECKPOINT TYPE field is set to LOGRECS, then this field must contain an integer from 1 000 through 16 000 000. This integer is the number of log records that are created between checkpoints, and the value is assigned to the CHKFREQ parameter. The value NOTUSED is assigned to the CHKLOGR parameter.

- If the CHECKPOINT TYPE field is set to MINUTES, then this field must contain the value NOTUSED. The CHKFREQ parameter is assigned the value of the MINUTES/CHECKPOINT field, and the CHKLOGR parameter is assigned the value NOTUSED.

- If the CHECKPOINT TYPE field is set to BOTH, then this field must contain an integer from 1 000 through 99 999 999. This integer is the number of log records that are created between checkpoints, and the value is assigned to the CHKLOGR parameter. The value NOTUSED is assigned to the CHKFREQ parameter.

- If the CHECKPOINT TYPE field is set to BOTH, Db2 prevents log checkpoints from being taken too frequently and degrading performance. Log checkpoints are scheduled, but not necessarily immediately taken, when the number of log records reaches the value of CHKLOGR.

**MINUTES/CHECKPOINT field (CHKFREQ and CHKMINS subsystem parameters)**

The MINUTES/CHECKPOINT field determines the approximate number of minutes that are to pass between scheduled log checkpoints.

- **Acceptable values:**
  - NOTUSED (if CHECKPOINT TYPE is LOGRECS);
  - 1 - 60 (if CHECKPOINT TYPE is MINUTES);
  - 1-1439 (if CHECKPOINT TYPE is BOTH)

**Default:** 5

**Update:** option 42 on panel DSNTIPB

**DSNZP:** DSN6SYS.P.CHKFREQ and DSN6SYS.P.CHKMINS

- If the CHECKPOINT TYPE field is set to LOGRECS, then this field must contain the value NOTUSED. The value of the RECORDS/CHECKPOINT field is assigned to the CHKFREQ parameter. The value NOTUSED is assigned to the CHKMINS parameter.
• If the CHECKPOINT TYPE field is set to MINUTES, then this field must contain an integer from 1 through 60. This integer is the number of minutes between log checkpoints, and the value is assigned to the CHKFREQ parameter. The value NOTUSED is assigned to the CHKMINS parameter.

• If the CHECKPOINT TYPE field is set to BOTH, then this field must contain an integer from 1 through 1439. This integer is the number of minutes between checkpoints, and the value is assigned to the CHKMINS parameter. The value NOTUSED is assigned to the CHKFREQ parameter.

• If the CHECKPOINT TYPE field is set to BOTH, Db2 prevents log checkpoints from being taken too frequently and degrading performance. Log checkpoints are scheduled, but not necessarily immediately taken, when the number of minutes that passes reaches the value of CHKMINS.

**UR CHECK FREQ field (URCHKTH subsystem parameter)**

Unless the option is disabled, Db2 issues a warning message to the console for an uncommitted unit of recovery (UR) after a certain number of checkpoint cycles. The URCHKTH subsystem parameter specifies the number of checkpoint cycles that are to complete before Db2 issues the warning message.

Acceptable values: 0 - 255
Default: 5
Update: option 42 on panel DSNTIPB
DSNZPxxx: DSN6SYSP URCHKTH

This option does not affect performance. If you use this option, specify a value that is based on how often a checkpoint occurs in your system and how much time you can allow for a restart or shutdown. For example, if your site's checkpoint interval is 5 minutes and the standard limit for issuing commits with units of recovery is 20 minutes, divide 20 by 5 to determine the best value for your system.

**UR LOG WRITE CHECK field (URLGWTH subsystem parameter)**

Unless this option is disabled, Db2 issues a warning message to the console after log records are written by an uncommitted unit of recovery (UR). The URLGWTH subsystem parameter specifies the number of log records that are to be written before Db2 issues a warning message.

Acceptable values: 0 - 1000K
Default: 10K
Update: option 42 on panel DSNTIPB
DSNZPxxx: DSN6SYSP URLGWTH

The purpose of this option is to provide notification of a long-running UR. Long-running URs might result in a lengthy Db2 restart or a lengthy recovery situation for critical tables. Specify the value in 1 K (1000 log records) increments. A value of 0 indicates that no write check is to be performed.

**LIMIT BACKOUT field (LBACKOUT subsystem parameter)**

The LBACKOUT subsystem parameter specifies whether Db2 postpones backward-log processing for some units of work.

Acceptable values: AUTO, YES, LIGHT, LIGHTAUTO, NO
Default: AUTO
Update: option 42 on panel DSNTIPB
DSNZPxxx: DSN6SYSP LBACKOUT
AUTO
Db2 postpones some backout processing, but automatically starts the backout processing when Db2 restarts and begins acceptance of new work. When Db2 is restarted with the LIGHT(YES) or LIGHT(NOINDOUBTS) option, Db2 ignores this setting and behaves as if LBACKOUT is set to NO.

YES
Db2 postpones backout processing for some units of work until the RECOVER POSTPONED command is issued. When Db2 is restarted with the LIGHT(YES) or LIGHT(NOINDOUBTS) option, Db2 ignores this setting and behaves as if LBACKOUT is set to NO.

LIGHT
Db2 postpones backout processing for some units of work until the RECOVER POSTPONED command is issued. When Db2 is restarted with the LIGHT(YES) or LIGHT(NOINDOUBTS) option, it creates but does not process postponed-abort units of recovery (URs). The URs are not processed until the RECOVER POSTPONED command is issued on a normal Db2 restart.

LIGHTAUTO
Db2 postpones some backout processing, but automatically starts the backout processing when Db2 restarts and begins acceptance of new work. When Db2 is restarted with the LIGHT(YES) or LIGHT(NOINDOUBTS) option, it creates but does not process postponed-abort URs. The postponed-abort URs are automatically processed after a normal Db2 restart.

NO
Specifies that Db2 backward-log processing should process all inflight and in-abort units of recovery (URs).

If you specify AUTO, YES, LIGHT, or LIGHTAUTO, backout processing runs concurrently with new work. Sets or partitions with pending backout work are unavailable until their backout work is complete.

Related concepts:
- Restart light (Db2 Data Sharing Planning and Administration)

Related tasks:
- Deferring restart processing (Db2 Administration Guide)

Related reference:
- START DB2 (Db2 Commands)
- BACKOUT DURATION field (BACKODUR subsystem parameter)

BACKOUT DURATION field (BACKODUR subsystem parameter)
The BACKODUR subsystem parameter is used to calculate the number of log records that are to be backed out during a restart. This value is applied when the LIMIT BACKOUT field is set to AUTO, YES, LIGHT, or LIGHTAUTO.

Acceptable values: 0 - 255
Default: 5
Update: option 42 on panel DSNTPPB
DSNZPxxx: DSN6SYS1 BACKODUR

During a restart, backward-log processing continues until the following events occur:
- All inflight and in-abort URs with update activity against the catalog or directory are backed out.
• All inflight and in-abort URs that only update temporary tables are backed out.

• The number of processed log records equals the number specified in the BACKOUT DURATION field multiplied by the number of log records per checkpoint. If the CHECKPOINT TYPE field is set to LOGRECS or BOTH, the value of the RECORDS/CHECKPOINT field is used as the number of log records per checkpoint. If the CHECKPOINT TYPE field is set to MINUTES, the default value of 500,000 is used as the number of log records per checkpoint.

Inflight and in-abort URs that are not completely backed out during restart are converted to postponed-abort status. Sets or partitions with postponed backout work are put into restart-pending (RESTP) state. This state blocks most access to the object. The exceptions are access by the RECOVER POSTPONED command state or by automatic backout processing that Db2 performs when the LIMIT BACKOUT field is set to AUTO or LIGHTAUTO.

A table space might be in restart-pending mode, without the associated index spaces also in restart-pending mode. This happens if a postponed-abort UR makes updates only to non-indexed fields of a table in a table space. In this case, the indexes are accessible to SQL (for index-only queries), even though the table space is inaccessible.

Related tasks:

- Deferring restart processing (Db2 Administration Guide)
- Resolving postponed units of recovery (Db2 Administration Guide)

Related reference:

- CHECKPOINT TYPE field (CHKTYPE subsystem parameter)
- LIMIT BACKOUT field (LBACKOUT subsystem parameter)
- RECOVER POSTPONED (Db2) (Db2 Commands)

Related information:

RO SWITCH CHKPTS field (PCLOSEN subsystem parameter)

The PCLOSEN subsystem parameter specifies the number of consecutive Db2 checkpoints that a page set or partition remains read-write after it is updated. After the specified number of checkpoints occur, Db2 converts the page set or partition to read-only.

Acceptable values: 1 - 32767
Default: 10 (checkpoints)
Update: option 42 on panel DSNTIPB
DSNZPxxx: DSN6SYS P CLOSEN

1 - 32767

The number of consecutive Db2 checkpoints that a page set or partition remains read-write after it is updated. After the specified number of checkpoints occur, Db2 converts the page set or partition from to read-only.

This value is used in conjunction with the value of the RO SWITCH TIME field. If the condition for either RO SWITCH TIME or RO SWITCH CHKPTS is met, the page set or partition is converted from read-write to read-only.
For NOT LOGGED table spaces, Db2 converts the page set or partition from read-write to read-only after one checkpoint, regardless of the value of RO SWITCH CHKPTS.

Having Db2 switch an infrequently updated set from read-write to read-only can be a performance benefit for recovery, logging, and for data sharing processing.

**Related tasks:**
- Testing group buffer pool caching

**Related reference:**
- RO SWITCH TIME field (PCLOSET subsystem parameter)

**RO SWITCH TIME field (PCLOSET subsystem parameter)**
The PCLOSET subsystem parameter specifies the number of minutes that a partition or page set remains read-write after it is updated. After the specified number of minutes, Db2 converts the set or partition to read-only.

Acceptable values: 1 - 32767
Default: 10 (minutes)
Update: option 42 on panel DSNTPB
DSNZP: DSN6SYSP PCLOSET

1 - 32767
The number of minutes that a page set or partition remains read-only after it is updated. After the specified number of minutes, Db2 converts the set or partition to read-only.

This value is used in conjunction with the value of the RO SWITCH CHKPTS field. If the condition for RO SWITCH CHKPTS or RO SWITCH TIME is met, the page set or partition is converted from read-write to read-only.

For NOT LOGGED table spaces, Db2 converts the page set or partition from read-write to read-only after one minute, regardless of the value of RO SWITCH TIME.

Having Db2 switch an infrequently updated set from read-write to read-only can be a performance benefit for recovery, logging, and for data sharing processing.

**Related tasks:**
- Testing group buffer pool caching

**Related reference:**
- RO SWITCH CHKPTS field (PCLOSEN subsystem parameter)

**LEVELID UPDATE FREQ field (DLDFREQ subsystem parameter)**
The DLDFREQ subsystem parameter specifies whether the level ID of a page set or partition is to be updated at Db2-determined checkpoint intervals.

Acceptable values: ON, OFF
Default: ON
Update: option 42 on panel DSNTPB
DSNZP: DSN6SYSP DLDFREQ

ON  Level ID updates for down-level detection are enabled.
OFF

Level ID updates for down-level detection are disabled.

**DSNTIPA: Archive log data set parameters panel**

The entries on the DSNTIPA panel define the characteristics of archive log data sets.

*Updating the parameters:* You can update all the parameters on this panel by using their subsystem parameter name.

---

**Figure 47. Archive log data sets panel: DSNTIPA**

**PRIMARY QUANTITY field (PRIQTY subsystem parameter)**

The PRIQTY subsystem parameter specifies the amount of primary space that is to be allocated for archive log data sets. The units for this parameter are specified in cylinders.

Acceptable values: 1 - 4369  
Default: 125  
Update: option 43 on panel DSNTIPB  
DSNZPxxx: DSN6ARVP PRIQTY  

For subsystems that are migrated from Db2 11, the PRIMARY QUANTITY field is determined according to the following rules:

*Table 33. PRIQTY values for subsystems that are migrated from Db2 11.*

<table>
<thead>
<tr>
<th>ALCUNIT at migration</th>
<th>How PRIQTY is determined in Db2 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLK</td>
<td>If the Db2 11 PRIQTY value is blank, the value is first calculated as in Db2 11 and then converted to cylinders. Otherwise, the Db2 11 PRIQTY value is converted to cylinders.</td>
</tr>
<tr>
<td>TRK</td>
<td>The Db2 11 PRIQTY value is converted to cylinders.</td>
</tr>
<tr>
<td>CYL</td>
<td>The Db2 11 PRIQTY value is used unchanged.</td>
</tr>
</tbody>
</table>
By default, DFSMS Direct Access Device Space Management (DADSM) limits the space allocation on a single volume to less than 65535 tracks. Therefore, if the archive log data set size can be greater than or equal to 65535 tracks, you need to specify a primary space quantity of less than 65535 tracks. This action forces the archive log data set to extend to a second volume.

Alternatively, the archive log data sets can be allocated by a DFSMS data class that has a DSNMTYP (data set name type) setting of LARGE or EXT. In this case, you can specify a primary space quantity larger than 65535 tracks without extending to a second volume. LARGE indicates that data sets in the data class are to be allocated in large physical sequential format. EXT indicates that data sets in the data class are to be allocated in extended physical sequential format. A setting of EXT is recommended, and it is required for striping of data sets. If you specify EXT, also set the IFEXT (if extended) parameter to R (required) rather than P (preferred).

**Related concepts:**

- Archive log data sets storage requirements

**SECONDARY QTY field (SECQTY subsystem parameter)**

The SECQTY subsystem parameter specifies the amount of secondary space that is to be allocated for a disk data set. The unit for this subsystem parameter value is cylinders.

Acceptable values: 1 - 4369
Default: 15
Update: option 43 on panel DSNTPBD
DSNZPxxx: DSN6ARVP SECQTY

For subsystems that are migrated from Db2 11, the PRIMARY QUANTITY field is determined according to the following rules:

<table>
<thead>
<tr>
<th>ALCUNIT at migration</th>
<th>How SECQTY is determined in Db2 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLK</td>
<td>If the Db2 11 SECQTY value is blank, the value is first calculated as in Db2 11 and then converted to cylinders. Otherwise, the Db2 11 SECQTY value is converted to cylinders.</td>
</tr>
<tr>
<td>TRK</td>
<td>The Db2 11 SECQTY value is converted to cylinders.</td>
</tr>
<tr>
<td>CYL</td>
<td>The Db2 11 SECQTY value is used unchanged.</td>
</tr>
</tbody>
</table>

**DEVICE TYPE 1 field (UNIT subsystem parameter)**

The UNIT subsystem parameter specifies the device type or unit name that is to be used for storing archive log data sets.

Acceptable values: device type or unit name
Default: TAPE
Update: option 43 on panel DSNTPBD
DSNZPxxx: DSN6ARVP UNIT

The value can be any alphanumeric string. If you choose to archive to disk, you can specify a generic device type with a limited volume range. Db2 requires all
archive log data sets that are allocated on disk to be cataloged. If the device type specifies disk, set the CATALOG DATA field on panel DSNTIPA to YES.

If the unit name specifies disk, the archive log data sets can extend to a maximum of 15 volumes. If the unit name specifies a tape device, Db2 can extend to a maximum of 20 volumes. If you chose to use disk, make the values of the PRIQTY and SECQTY subsystem parameters large enough to contain all of the data that comes from the active log data sets without extending beyond 15 volumes.

When archiving to disk, Db2 uses the number of online storage volumes for the specified UNIT name to determine a count of candidate volumes, up to a maximum of 15 volumes. If the archives are to be managed by SMS, do not use a storage class with the Guaranteed Space attribute. SMS attempts to allocate a primary extent on every candidate volume. This can result in allocation failures or unused space because the primary extent on each unused volume is not released when the archive data set is closed.

Related concepts:

Archive log data sets (Db2 Administration Guide)

Related reference:

PRIMARY QUANTITY field (PRIQTY subsystem parameter)
SECONDARY QTY field (SECQTY subsystem parameter)

Related information:

DEVICE TYPE 2 field (UNIT2 subsystem parameter)
The UNIT2 subsystem parameter specifies the device type or unit name that is to be used for storing the second copy of archive log data sets. These data sets are the COPY2 data sets.

Acceptable values: device type or unit name
Default: none
Update: option 43 on panel DSNTIPB
DSNZPxxx: DSN6ARVP UNIT2

BLOCK SIZE field (BLKSIZE subsystem parameter)
The BLKSIZE subsystem parameter determines the block size that is to be used for the archive log data set.

Acceptable values: 8192 - 28672
Default: 24576
Update: option 43 on panel DSNTIPB
DSNZPxxx: DSN6ARVP BLKSIZE

The block size must be compatible with the device type that you use for archive logs. The value is rounded up to the next multiple of 4096 bytes. You can also enter the value with a K; for example, 28K.

If the archive log is written to tape, use the largest possible block size to improve the speed of reading the archive logs. Use the table below as a guide.
### Table 35. Recommended block size values

<table>
<thead>
<tr>
<th>Archive log device</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tape</td>
<td>28672</td>
</tr>
<tr>
<td>3390 geometry disk</td>
<td>24576</td>
</tr>
</tbody>
</table>

**READ TAPE UNITS field (MAXRTU subsystem parameter)**

The MAXRTU subsystem parameter specifies the maximum number of dedicated tape units that can be allocated to concurrently read archive log tape volumes.

- **Acceptable values:** 1 - 99
- **Default:** 2
- **Update:** option 43 on panel DSNTPB
- **DSNZPxxx:** DSN6LOGP MAXRTU

This installation option, along with the DEALLOC PERIOD field, allows Db2 to optimize archive log reading from tape devices.

In a data sharing environment, the archive tape is not available to other members of the group until the deallocation period expires. You might not want to use this option in a data sharing environment unless all recover jobs are submitted from the same member.

**Recommendation:** Set the READ TAPE UNITS value to be at least one less than the number of tape units available to Db2. If you do otherwise, the OFFLOAD process could be delayed, which would affect the performance of your Db2 subsystem. For maximum throughput during archive log processing, specify the largest value possible for this option, remembering that you need at least one tape unit for offload processing. You can override this value by using the SET ARCHIVE command.

**DEALLOC PERIOD field (DEALLCT subsystem parameter)**

The DEALLCT subsystem parameter determines the length of time that an archive read tape unit is to be allowed to remain unused before it is deallocated.

- **Acceptable values:** minutes,seconds, 1440, NOLIMIT
- **Default:** 0
- **Update:** option 43 on panel DSNTPB
- **DSNZPxxx:** DSN6LOGP DEALLCT

- **minutes,seconds**
  - Sets the amount of time before deallocation as the specified number of minutes (0 – 1440) and number of seconds (0 – 59). Separate the number of minutes and number of seconds with a comma and no spaces.

- **1440**
  - Sets the amount of time before deallocation to 1440 minutes.

- **NOLIMIT**
  - Sets no limit on the amount of time before deallocation. Specifying NOLIMIT allows maximum optimization opportunities.

**Recommendation:** If you are not using data sharing and your archive log data is on tape, set this value high enough to allow Db2 to optimize tape handling for multiple read applications. When all tape reading is complete, you can update this option with the SET ARCHIVE command. If you are using data sharing, tapes
should be deallocated quickly to avoid deadlocks. If Db2 is running in data sharing mode, it is strongly recommended that you use disk archive logs, not tape archive logs.

**Examples**

Assume that you want the deallocation period to be 30 seconds. Enter 0,30.

Assume that you want the deallocation period to be 23 minutes and 47 seconds. Enter 23,47.

**RECORDING MAX field (MAXARCH subsystem parameter)**

The MAXARCH subsystem parameter controls the maximum number of archive log volumes that are to be recorded in the BSDS.

Acceptable values: 10 - 10000
Default: 10000
Update: option 43 on panel DSNTIPB
DSNZPxxx: DSN6LOGP MAXARCH

When this number is exceeded, recording resumes at the beginning of the BSDS.

You must create image copies of all Db2 objects, probably several times, before the archive log data sets are discarded. If you fail to retain an adequate number of archive log data sets for all the image copies, you might need to cold start or reinstall Db2. In both cases, data is lost.

**Related tasks:**

[Managing the log (Db2 Administration Guide)]

**WRITE TO OPER field (ARCWTOR subsystem parameter)**

The ARCWTOR subsystem parameter controls whether Db2 is to send a message to the operator and wait for an answer before attempting to mount an archive log data set.

Acceptable values: YES, NO
Default: YES
Update: option 43 on panel DSNTIPB
DSNZPxxx: DSN6ARVP ARCWTOR

Other Db2 users can be forced to wait while the mount is pending. They are not affected while Db2 is waiting for a response to the message.

**YES** Specifies that the system is to send a message to the operator before attempting to mount an archive log data set. Specify this value if you use a device for storing archive log data sets, such as tape, that requires long delays for mounts.

**NO** Specifies that the system will not send a message to the operator and wait for an answer before attempting to mount an archive log data set. Use this value if you use a device that does not have long delays for mounts.

The DEVICE TYPE 1 field specifies the device type or unit name.
**WTOR ROUTE CODE field (ARCWRTC subsystem parameter)**

The ARCWRTC subsystem parameter identifies the list of route codes to be used in messages from the archive log data sets to the operator.

Acceptable values: 1 - 16 route codes. Specify at least one code. Separate multiple codes by commas only, with no blanks; for example: 1,3,5,7,9,10,11

Default: 1,3,4

Update: option 43 on panel DSNTIPB

DSNZPxxx: DSN6ARVP ARCWRTC

Related reference:
- [Routing Codes (MVS System Messages)](Routing Codes (MVS System Messages))
- [Routing the Message (z/OS MVS Assembler Services Guide)](Routing the Message (z/OS MVS Assembler Services Guide)

**RETENTION PERIOD field (ARCRETN subsystem parameter)**

The ARCRETN subsystem parameter specifies the retention period, which is the number of days that Db2 is to retain archive log data sets. The retention period is added to the current date to calculate the expiration date of the archive log data sets.

Acceptable values: 0 - 9999

Default: 9999

Update: option 43 on panel DSNTIPB

DSNZPxxx: DSN6ARVP ARCRETN

The retention period is often used in tape management systems to control the reuse and scratching of data sets and tapes. Db2 uses the ARCRETN value as the value for the dynamic allocation parameter DALRETPD when archive log data sets are created.

The retention period set by the DFSMSdfp storage management subsystem (SMS) can be overridden by this Db2 parameter. Typically, the retention period is set to the smaller value that is specified by either Db2 or SMS. The storage administrator and database administrator should agree on a retention period value that is appropriate for Db2.

**Important:** A wide variety of tape management systems exists, along with the opportunity for external manual overrides of retention periods. Consequently, Db2 does not have an automated method to delete the archive log data sets from the BSDS inventory of archive log data sets. Therefore, the information about an archive log data set might be in the BSDS after its retention period has expired. The information might be held long after the tapes have been scratched by a tape management system. Conversely, the maximum number of archive log data sets might be exceeded (see READ TAPE UNITS field). In such a case, the data from the BSDS might be dropped long before the data set reaches its expiration date.

Related reference:
- [DSNJU003 (change log inventory) (Db2 Utilities)](DSNJU003 (change log inventory) (Db2 Utilities))

**QUIESCE PERIOD field (QUIESCE subsystem parameter)**

The QUIESCE subsystem parameter determines the maximum amount of time, in seconds, that Db2 is to be allowed to attempt a full system quiesce.

Acceptable values: 1 - 999

---

Installation and Migration Guide
**Default:** 5  
**Update:** option 43 on panel DSNTIPB  
**DSNZPxxx:** DSN6ARVP QUIESCE

This parameter requires some tuning. If you specify too short an interval, the quiesce period expires before a full quiesce is accomplished. If you specify too long an interval, the quiesce period might cause unnecessary Db2 lock contention and timeouts.

**Related tasks:**

[Quiescing activity before offloading (Db2 Administration Guide)](#)

**COMPACT DATA field (COMPACT subsystem parameter)**

The COMPACT subsystem parameter controls whether data that is written to archive logs is to be compacted.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>YES or NO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default:</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>Update:</strong></td>
<td>option 43 on panel DSNTIPB</td>
</tr>
<tr>
<td><strong>DSNZPxxx:</strong></td>
<td>DSN6ARVP COMPACT</td>
</tr>
</tbody>
</table>

This option only applies to data that is written to a tape device that has the improved data recording capability (IDRC) feature. When this feature is turned on, hardware in the IDRC-compliant tape control unit uses several techniques to increase the amount of data that can be written on a tape. Specify NO if you do not want your data to be compacted or you do not use a tape device with the IDRC feature. Specify YES if you use a tape device with the IDRC feature and you want the data to be compacted.

**Recommendation:** Archive to disk and then use another facility, such as DFSMSshm, to archive the archive log from disk to tape. Be aware that data that is compressed to tape can only be read with a device that supports the IDRC feature. This could be a concern when you send archive tapes to another site for remote recovery.

**SINGLE VOLUME field (SVOLARC subsystem parameter)**

The SVOLARC subsystem parameter specifies whether Db2 is to use a single volume for disk archives.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>YES, NO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default:</strong></td>
<td>NO</td>
</tr>
<tr>
<td><strong>Update:</strong></td>
<td>option 43 on panel DSNTIPB</td>
</tr>
<tr>
<td><strong>DSNZPxxx:</strong></td>
<td>DSN6ARVP SVOLARC</td>
</tr>
</tbody>
</table>

**NO** When archiving to disk, Db2 determines a count of candidate volumes, up to 15, based on the number of online volumes that have a use attribute of STORAGE for the specified UNIT name. If one or zero volumes are mounted with STORAGE as the use attribute, Db2 does not pass a volume or unit count, which is effectively the same as specifying SVOLARC=YES.

**YES** Db2 specifies no unit or volume count. The result is the default unit and volume count of 1, unless SMS overrides it. Use YES for SMS-managed data sets. SMS provides the volume count from the data class. If no value is specified in the data class or no data class is assigned the default volume count is 1.
Related concepts:

Tips for archiving with z/OS DFSMS (Db2 Administration Guide)

DSNTPS: Databases and spaces to start automatically panel

The entries on the DSNTPS panel name the databases, table spaces, and index spaces to restart automatically when you start Db2.

Updating the parameters: You can update all parameters on this panel by using their subsystem parameter name.

![DSNTPS panel](image)

Figure 48. Databases and spaces to start automatically panel: DSNTPS

**RESTART OR DEFER field (RESTART subsystem parameter)**

The RESTART subsystem parameter determines whether Db2 is to restart or defer processing for the specified databases, table spaces, and index spaces when Db2 is started. The specified objects are those that are listed in fields 2 through 37 of panel DSNTPS

Acceptable values: RESTART, DEFER

Default: RESTART

Update: option 44 on panel DSNTPIB

DSNZIPXXX: DSN6SPRM RESTART

**RESTART**

Causes Db2 to perform restart processing for the listed objects.

**DEFER**

Causes Db2 not to perform restart processing for the objects.

**START NAMES field (ALL subsystem parameter)**

The ALL subsystem parameter specifies the databases, table spaces, and index spaces that are to be restarted or deferred when Db2 starts. The value of the RESTART OR DEFER field determines whether the object is to be restarted or if the restart is to be deferred.
Acceptable values: ALL, space names
Default: ALL
Update: option 44 on panel DSNTIPB
DSNZPxxx: DSN6SPRM ALL

ALL

The restart or defer restart applies to all Db2 databases, table spaces, and index spaces. If you use this value, leave the remaining fields on the DSNTIPS panel blank. Use ALL in this field and DEFER in the RESTART OR DEFER field to defer recovery of all objects, including Db2 catalog objects. Trusted connections are not available if ALL is used in this field and DEFER is used in the RESTART OR DEFER field.

ALL is the default value.

<database name>
The restart or defer restart applies to all spaces in the database.

<table space or index space name> in the format database-name.space-name
The restart or defer restart applies to the individual table space or index space.

You can specify up to 36 object names on this panel. If you want to control restart processing for more than 36 objects, edit job DSNTIJUZ after you run the CLIST, and add the object names as ending positional parameters to macro DSN6SPRM. You can add up to 2500 object names in DSNTIJUZ.

Related tasks:
Deferring restart processing (Db2 Administration Guide)

DSNTIPR: Distributed data facility panel 1
The entries on the DSNTIPR panel control the starting of the distributed data facility (DDF). The entries on the DSNTIPR panel also specify location names that are used to connect another Db2 subsystem.

Figure 49. Distributed data facility panel: DSNTIPR
Related concepts:

Support for a communications network

DDF STARTUP OPTION field (DDF subsystem parameter)
The DDF subsystem parameter controls whether DDF is to be loaded at startup and, if DDF is to be loaded, how it is to be started.

Acceptable values: NO, AUTO, COMMAND
Default: NO
Update: option 45 on panel DSNTIPB

NO Signifies that DDF is not to be loaded at Db2 startup and that it cannot be started with a command. If you specify NO, the stored procedures sample application and DDF sample jobs (DSNTEJ6, DSNTEJ6D, DSNTEJ6P, DSNTEJ6S, DSNTEJ6T, DSNTEJ6Z, DSNTEJ61, DSNTEJ62, DSNTEJ63, DSNTEJ64, DSNTEJ65, and DSNTEJ66) are not edited. Also, the remaining fields on the DSNTIPR panel are ignored. However, the remaining fields require entries, so you can accept the default values.

AUTO Specifies that this facility is to be automatically initialized and started when the Db2 subsystem is started.

COMMAND Specifies that the facility is to be initialized at Db2 startup and is prepared to receive the DSN1 START DDF command.

If AUTO or COMMAND is specified, the DDF address space is started as part of DDF initialization. The remaining fields on the DSNTIPR panel are mandatory. The repository for the field names (LOCATION, LUNAME, and PASSWORD) is the bootstrap data set (BSDS). The BSDS is updated by the change log inventory utility in step DSNTLOG of installation job DSNTIJUZ.

DB2 LOCATION NAME field
The value of the DB2 LOCATION NAME field specifies the unique name that requesters use to connect to this Db2 subsystem. The value of the DB2 LOCATION NAME field must be the same on all members within a Db2 data sharing group.

Acceptable values: 1 - 16 alphanumeric characters, beginning with a letter, and containing no special characters
Default: LOC1
Update: see "Updating subsystem parameters that cannot be updated by using DSNTIPB" on page 373

DB2 VTAM LUNAME field
The value of the DB2 VTAM LUNAME field specifies the LU name that is to uniquely identify this Db2 subsystem to VTAM. The name is also to be used to uniquely identify logical units of work within Db2 trace records.
Acceptable values: 1 - 8 alphanumeric characters, beginning with a letter and containing no special characters
Default: none
Update: see Updating subsystem parameters that cannot be updated by using DSNTIPB
DSNZPxxx: none

You must specify a value for this field.

The DB2 VTAM LUNAME setting identifies this Db2 subsystem within your network, identifies transactions for indoubt resolution, and correlates trace records. The LUNAME value must be unique for each member in a Db2 data sharing group. It is used to start SNA communications. TCP/IP communications are also started if TCP/IP PORT or SECURE PORT, or both, and RESYNC PORT are specified. To start TCP/IP communication only, specify DB2 TCP/IP IPNAME with TCP/IP PORT or SECURE PORT, or both, and specify RESYNC PORT.

**DB2 VTAM PASSWORD field**

The optional DB2 VTAM PASSWORD field specifies the password that VTAM is to use to recognize this Db2 subsystem.

Acceptable values: 1 - 8 alphanumeric characters
Default: none
Update: see Update on Updating subsystem parameters that cannot be updated by using DSNTIPB
DSNZPxxx: none

This password must also be supplied to VTAM on the VTAM APPL definition statement. The password must begin with a letter and must not contain special characters.

**DB2 GENERIC LUNAME field**

The value of the DB2 GENERIC LUNAME field specifies a generic LU name that identifies this Db2 subsystem or data sharing group in a network. The DB2 GENERIC LUNAME field value must be the same for all members in a Db2 data sharing group.

Acceptable values: 1 - 8 alphanumeric characters
Default: none
DSNZPxxx: none

You can use a generic LU name only if Db2 is running as part of a z/OS Sysplex. Using a generic LU name helps you control the distributed workload among the servers in a data sharing group. Previously, you could associate only one LU name with a LOCATION name. Now, you can associate multiple NETID.LUNAME values with a single LOCATION name. When an application requests access to a particular location, Db2 uses the SYSIBM.LOCATIONS and SYSIBM.LULIST tables to find the available network destinations (LU names) for that location.

Related tasks:

- Connecting distributed database systems
- Configuring data sharing groups for group-generic access (Db2 Data Sharing Planning and Administration)
DB2 TCP/IP IPNAME field
The value of the DB2 TCP/IP IPNAME field identifies and associates an IPNAME value with DDF.

Acceptable values: 1 - 8 alphanumeric characters, beginning with a letter or alphabet extender

Default: See Table 36
Update: option 45 on panel DSNTIPB
DSNZPxxx: none

The DB2 TCP/IP IPNAME identifies transactions for indoubt resolution, correlates trace records and is the Db2 group RACF PassTicket APPLID. The IPNAME value must be the same for each member in a Db2 data sharing group. If set, TCP/IP communications is started, but not SNA communications, regardless if there is a value for LUNAME. Inbound and outbound communications over TCP/IP are allowed. Db2 makes no attempt to activate VTAM communications support.

Table 36. Initial values for DB2 TCP/IP IPNAME field on DSNTIPR

<table>
<thead>
<tr>
<th>INSTALL TYPE</th>
<th>Data sharing?</th>
<th>IPNAME updateable?</th>
<th>Initial value if DSNTIDXI IPNAME parameter is not NULL</th>
<th>Initial value if DSNTIDXI IPNAME parameter is NULL or not found</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTALL</td>
<td>No</td>
<td>Yes</td>
<td>SUBSYSTEM NAME field on panel DSNTIPM, if it is modified</td>
<td>BLANK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Otherwise, DSNTIDXI IPNAME parameter</td>
<td></td>
</tr>
<tr>
<td>INSTALL</td>
<td>First member</td>
<td>Yes</td>
<td>GROUP NAME field on panel DSNTIPK</td>
<td>BLANK</td>
</tr>
<tr>
<td>INSTALL</td>
<td>Subsequent member</td>
<td>No</td>
<td>DSNTIDXI IPNAME parameter (from first data sharing group member)</td>
<td>BLANK</td>
</tr>
<tr>
<td>INSTALL</td>
<td>Enable</td>
<td>Yes</td>
<td></td>
<td>BLANK</td>
</tr>
<tr>
<td>MIGRATE</td>
<td>No</td>
<td>No</td>
<td>BLANK</td>
<td>BLANK</td>
</tr>
<tr>
<td>MIGRATE</td>
<td>Yes</td>
<td>No</td>
<td>BLANK</td>
<td>BLANK</td>
</tr>
<tr>
<td>UPDATE</td>
<td>N/A</td>
<td>Yes</td>
<td>DSNTIDXI IPNAME parameter</td>
<td>BLANK</td>
</tr>
</tbody>
</table>

Related concepts:
- Specifying a generic LU name and IPNAME value for the data sharing group (Db2 Data Sharing Planning and Administration)
- Member-specific access (Db2 Data Sharing Planning and Administration)

Related reference:
TCP/IP PORT field
The value of the TCP/IP PORT field determines the TCP/IP port number that is to be used for accepting TCP/IP connection requests from remote DRDA clients and REST services.

Acceptable values: 1 - 65534
Default: 446
Update: option 45 on panel DSNTIPB
DSNZPxxx: none

If you are enabling data sharing, each member must have the same TCP/IP port number. You must specify a value for the TCP/IP PORT field if you plan to use TCP/IP. Leaving this field blank means that you are not using TCP/IP. A blank field is equivalent to using 0 in the change log inventory (DSNJU003) utility.

Related reference:

SECURE PORT field
The value of the SECURE PORT field specifies the TCP/IP port number that is to be used for accepting secure TCP/IP connection requests from remote DRDA clients and REST services.

Acceptable values: 1 - 65534
Default: none
Update: option 45 on panel DSNTIPB
DSNZPxxx: none

If you are enabling data sharing, each member must have the same secure TCP/IP port number. You must specify a value if you plan to use TCP/IP with Secure Socket Layer (SSL). Leaving this field blank means that you are not using TCP/IP with SSL. A blank field is equivalent to using 0 in the change log inventory (DSNJU003) utility.

Related reference:

RESYNC PORT field
The value of the RESYNC PORT field specifies the TCP/IP port number that is to be used to process requests for two-phase commit resynchronization.

Acceptable values: 1 - 65534
Default: 5001
Update: option 45 on panel DSNTIPB
DSNZPxxx: none

This value must be different than the values that are specified for TCP/IP PORT and SECURE PORT. If you are enabling data sharing, each member must have a unique resynchronization port. Leaving this field blank means that you are not using TCP/IP. A blank field is equivalent to using 0 in the change log inventory (DSNJU003) utility.

Related reference:
RESYNC INTERVAL field (RESYNC subsystem parameter)

A resynchronization period is the time during which indoubt logical units of work that involve this Db2 subsystem and partner logical units are processed. The RESYNC subsystem parameter specifies the time interval, in minutes, that is to occur between resynchronization periods.

Acceptable values: 1 - 99
Default: 2
Update: option 45 on panel DSNTIPB
DSNZPxxx: DSN6FAC RESYNC

DSNTIP5: Distributed data facility panel 2

The entries on the DSNTIP5 panel are used to configure the distributed data facility (DDF).

DDF THREADS field (CMTSTAT subsystem parameter)

The CMTSTAT subsystem parameter controls whether threads are to be made active or inactive after they successfully commit or roll back and hold no cursors. A thread can become inactive only if it holds no cursors, has no temporary tables defined, and executes no statements from the dynamic statement cache.

Acceptable values: ACTIVE, INACTIVE
Default: INACTIVE
Update: option 46 on panel DSNTIPB
DSNZPxxx: DSN6FAC CMTSTAT

ACTIVE

Threads remain active after they successfully commit or roll back and hold no
cursors. This setting provides the best performance but consumes system resources. If your installation must support a large number of connections, specify INACTIVE instead.

INACTIVE

Db2 supports two different types of inactive concepts:

- An inactive DBAT (previously called a type 1 inactive thread), which has the same characteristics as inactive threads that were available in releases prior to Db2 12. In this case, the thread remains associated with the connections, but the thread’s storage utilization is reduced as much as possible. However, this still potentially requires a large number of threads to support a large number of connections.

- An inactive connection (previously called a type 2 inactive thread), which uses less storage than an inactive DBAT. In this case, the connections are disassociated from the thread. The thread is allowed to be pooled and reused for other connections, new or inactive. This provides better resource utilization because there are typically a small number of threads that can be used to service a large number of connections.

Because they use less storage, inactive connections are preferable. However, not all threads can become inactive connections. The following table summarizes the conditions under which you can have an inactive DBAT or an inactive connection. If a thread is to become inactive, Db2 tries to make it an inactive connection. If Db2 cannot make it an inactive connection, it tries to make it a inactive DBAT. If neither attempt is successful, the thread remains active.

**Table 37. Requirements for inactive DBATs and inactive connections**

<table>
<thead>
<tr>
<th>If the event is...</th>
<th>Inactive connection can be created</th>
<th>Inactive DBAT can be created</th>
</tr>
</thead>
<tbody>
<tr>
<td>A hop to another location with a pending request¹</td>
<td>No</td>
<td>Yes²</td>
</tr>
<tr>
<td>A package bound with RELEASE(COMMIT)</td>
<td>Yes</td>
<td>Yes²</td>
</tr>
<tr>
<td>A package bound with RELEASE(DEALLOCATE)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>An outstanding open and held cursor³, a held LOB locator, or a package bound with KEEP_DYNAMIC(YES)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>A declared temporary table that is active (the table was not explicitly dropped through the DROP TABLE statement or the ON COMMIT DROP TABLE clause on the DECLARE GLOBAL TEMPORARY TABLE statement)</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Notes:

1. A pending request occurs when a hop connection is created but no SQL statements are run at that location.

2. An inactive DBAT can only be created if the MAXTYPE1 subsystem parameter is set to a value greater than 0.

3. A cursor can be closed with fast implicit close. For more information, see [Fast implicit close (Db2 Performance)](#).

IBM Systems Virtualization Engine Enterprise Workload Manager™ (EWLM) is a robust performance management tool that allows you to monitor and manage...
work that runs within your environment. If you use EWLM with Db2, specify
INACTIVE for DDF THREADS. Otherwise, DDF ignores all EWLM correlators that
are passed to it.

**MAX INACTIVE DBATS field (MAXTYPE1 subsystem parameter)**
The MAXTYPE1 subsystem parameter determines the number of inactive DBATs
that Db2 is to allow.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>0 to the value of the MAX REMOTE CONNECTED field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>0</td>
</tr>
<tr>
<td>Update:</td>
<td>option 46 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6FAC MAXTYPE1</td>
</tr>
</tbody>
</table>

This limit is defined because a large number of inactive DBATs might adversely
affect system performance. Inactive DBATs were traditionally used for private
protocol, while DRDA uses inactive connections. However, inactive DBATs can
occur in DRDA when Db2 acts as an intermediate server, receiving a connection
from a client, and the thread then hops to another location while a held cursor
spans the commit scope.

A value of 0 indicates that inactive DBATs are not allowed. If a thread meets the
requirement of an inactive DBATs, and MAX INACTIVE DBATS is 0, the thread
remains active.

A value of greater than 0 indicates that inactive DBATs are allowed, but they are
limited to the specified number. When a thread meets the requirement of an
inactive DBAT, and MAX INACTIVE DBATS is reached, the remote connection is
terminated.

If you want to allow inactive DBATs, set this value to the maximum number of
concurrent connections that you want to allow to go inactive that access another
remote location with three-part names.

A value that is equal to the value in the MAX REMOTE CONNECTED field from
panel DSNTIPE allows all remote threads to become type 1 inactive threads.

**IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter)**
The IDTHTOIN subsystem parameter controls the amount of time, in seconds, that
an active server thread is to be allowed to remain idle. This amount of time is the
timeout value. The thread is canceled after the timeout value expires, and its locks
and cursors are released.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>0 - 9999</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>120</td>
</tr>
<tr>
<td>Update:</td>
<td>option 46 on panel DSNTIPB</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>DSN6FAC IDTHTOIN</td>
</tr>
</tbody>
</table>

Inactive and indoubt threads are not subject to timeout. The value that you specify
for the DDF THREADS field determines whether a thread can become inactive,
and thus not subject to timeout.

Threads are checked every two minutes to see if they have exceeded the timeout
value. If the timeout value is less than two minutes, the thread might not be
canceled if it has been inactive for more than the timeout value but less than two
minutes.
Specifying 0 disables timeout processing. If timeout processing is disabled, idle server threads remain in the system and continue to hold any resources.

**EXTENDED SECURITY field (EXTSEC subsystem parameter)**
The EXTSEC subsystem parameter specifies how two related security options are to be set. These settings control what happens when a DDF connection has security errors and whether RACF users can change their passwords through the DRDA change password function.

- **Acceptable values:** YES, NO
- **Default:** YES
- **Update:** option 46 on panel DSNTPBP
- **DSNZPxxx:** DSN6SYSP EXTSEC
- **Subsystem parameter:** Yes

**YES**
Detailed reason codes are returned to a DRDA level 3 client when a DDF connection request fails because of security errors. When using SNA protocols, the requester must have included a product that supports the extended security sense codes. One such product is Db2 Connect.

RACF users can change their passwords by using the DRDA change password function. This support is only for DRDA requesters that have implemented support for changing passwords.

**NO**
Generic error codes are returned to the clients and RACF users are prevented from changing their passwords.

**Recommendation:** Specify a value of YES. This setting allows properly enabled DRDA clients to determine the cause of security failures without requiring Db2 operator support. A value of YES also allows RACF users on properly enabled Db2 clients to change their passwords.

**Note:** This is a security-related parameter. When this parameter is set to YES, detailed reason codes are returned to the client when a DDF connection request fails because of security errors that might enable more malicious attacks. If this parameter is set to YES, RACF users can change their passwords by using the DRDA change password function.

**TCP/IP ALREADY VERIFIED field (TCPALVER subsystem parameter)**
The TCPALVER subsystem parameter specifies whether Db2 is to accept TCP/IP connection requests containing only a user ID, or if a stronger form of security is required. If Db2 is to accept only a user ID, no password, RACF PassTicket, or Kerberos ticket is needed.

The TCPALVER subsystem parameter has no effect in the following situations:
- Trusted context users that have been switched
- Connections that use SECPOR, including SSL implementations, such as AT-TLS

- **Acceptable values:** YES, CLIENT, NO, SERVER, SERVER_ENCRYPT
- **Default:** NO (SERVER)
- **Update:** option 46 on panel DSNTPBP
- **DSNZPxxx:** DSN6FAC TCPALVER
- **Security parameter:** Yes
YES  A new connection is accepted with a user ID only.

CLIENT
This value can be used as an alternative to YES.

NO  A user ID and password are required for connection requests, or the connection must be authenticated by a RACF PassTicket or Kerberos ticket. The user ID and password can be encrypted or non-encrypted.

SERVER
This value can be used as an alternative to NO.

SERVER_ENCRYPT
A user ID and password are required for connection requests. Kerberos tickets are also accepted. In addition, one of the following must be true:

- The user ID and password is AES (Advanced Encryption Standard)-encrypted.
- The connection is accepted on a port that ensures AT-TLS (Application Transparent - Transport Layer Security) policy protection, such as a Db2 security port (SECPORT).

Non-encrypted security credentials are not accepted unless the connection is secured by the TCP/IP network. DES-based (Data Encryption Standard) encryption is also considered insecure. If RACF PassTickets are used, Db2 must be permitted to execute the SAF IRRSPK00 service. That is, the use of SERVER_ENCRYPT requires a RACF PERMIT for Db2 (the user ID associated to the ssnmDIST started task ID) to execute the IRRSPK00 service, as shown in the following exampt RACF commands:

```
RDEF PTKTDATA IRRPTAUTH.SYEC1B.* UACC(NONE)
PE IRRPTAUTH.SYEC1B.* CLASS(PTKTDATA) ID(SYSDSP) ACCESS(READ)
SETR RACLIST(PTKTDATA) REFRESH
```

This value must be the same for all members of a data sharing group. This option applies to all incoming requests that use TCP/IP, regardless of the requesting location.

Attention: This is a security-related parameter. A setting of YES or CLIENT provides minimal security. With YES or CLIENT, Db2 is subject to security attacks, because the identity of the user or process that is attempting to gain access is not verified. Specify these settings only if you have a highly secure network.

Recommendation: Setting the parameter to SERVER_ENCRYPT provides the best security. Connections are accepted only if user credentials are provided to authenticate the user ID, and strong encryption is used to protect the user ID and credentials.

Related tasks:

[Managing inbound TCP/IP-based connection requests (Managing Security)](#)

**EXTRA BLOCKS REQ field (EXTRAREQ subsystem parameter)**
The EXTRAREQ subsystem parameter limits the number of extra DRDA query blocks that Db2 is to request from a remote DRDA server. This parameter does not limit the size of the SQL query answer set. It simply controls the total amount of data that can be transmitted on a network exchange.

- Acceptable values: 0 - 100
- Default: 100
- Update: option 46 on panel DSNTIPB
EXTRA BLOCKS SRV field (EXTRASRV subsystem parameter)
The EXTRASRV subsystem parameter limits the number of extra DRDA query blocks that Db2 is to return to a DRDA client. This parameter not limit the size of the SQL query answer set. It simply controls the total amount of data that can be transmitted on a network exchange.

Acceptable values: 0 - 100
Default: 100
Update: option 46 on panel DSNTIPB

TCP/IP KEEPALIVE field (TCPKPALV subsystem parameter)
The TCPKPALV subsystem parameter determines whether the TCP/IP KeepAlive value in the TCP/IP configuration is to be overridden with a different value. This option can be used in cases where the TCP/IP KeepAlive value in the TCP/IP configuration is not appropriate for the Db2 subsystem.

Acceptable values: ENABLE, DISABLE, or 1 to 65534
Default: 120
Update: option 46 on panel DSNTIPB

ENABLE
The TCP/IP KeepAlive configuration value is not overridden.

DISABLE
KeepAlive probing is disabled for this subsystem.

1 - 65534
The TCP/IP KeepAlive configuration value is overridden with the value of the TCP/IP KEEPALIVE field. This value is specified in seconds. Consider setting this value close to the IDLE THREAD TIMEOUT value on installation panel DSNTIPR or the RESOURCE TIMEOUT value on installation panel DSNTIP1.

Avoid using very small values. KeepAlive detection is accomplished by probing the network based on the time that is entered in the KeepAlive parameter. A small KeepAlive value can cause excessive network traffic and system resource consumption. Maintain a proper balance that allows network failures to be detected on a timely basis without a severe impact on system and network performance.

Related tasks:
Timing out idle active threads (Db2 Performance)

POOL THREAD TIMEOUT field (POOLINAC subsystem parameter)
The POOLINAC subsystem parameter specifies the approximate time, in seconds, that a database access thread (DBAT) can remain idle in the pool before it is terminated.

Acceptable values: 0 - 9999
Default: 120
Update: option 46 on panel DSNTIPB

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST
A database access thread in the pool counts as an active thread against MAX
REMOTE ACTIVE and can hold locks, but does not have any cursors.

Threads are checked every two minutes to see if they have exceeded the timeout
value. If the timeout value is less than two minutes, the thread might not be
canceled if it has been inactive for more than the timeout value but less than two
minutes.

Specifying 0 causes a DBAT to terminate rather than go into the pool if the pool
has a sufficient number of threads to process the number of inactive DBATs (type 2
inactive threads) that currently exist.

**CONN QUEUE MAX DEPTH field (MAXCONQN subsystem parameter)**
The MAXCONQN subsystem parameter specifies the maximum number of inactive
or new connections that can be queued waiting for a DBAT to process the request.

Acceptable values: OFF, ON, 1 - 19999
Default: OFF
Update: option 46 on panel DSNTIPB
DSNZPxxx: DSN6FAC MAXCONQN

**OFF**
The depth of the connection queue is limited by the value of the CONDBAT
subsystem parameter.

**ON**
The depth of the connection queue is limited by the value of the MAXDBAT
subsystem parameter.

**Numeric value**
The maximum number of connections that can be queued waiting for a DBAT
to process a request.

When a request is added to the connection request queue and the thresholds
specified by both the MAXDBAT and MAXCONQN subsystem parameters are
both reached (unless MAXCONQN is set to OFF) then DDF closes the client
connection longest waiting client connection in the queue. The closed connections
give remote clients an opportunity to redirect the work to other members of the
group that have more resources to process the work. The function is enabled only
when Db2 subsystem is a member of a data sharing group.

This subsystem parameter can be updated online.

Related reference:
MAX REMOTE ACTIVE field (MAXDBAT subsystem parameter)
MAX REMOTE CONNECTED field (CONDBAT subsystem parameter)

**CONN QUEUE MAX WAIT field (MAXCONQW subsystem parameter)**
The MAXCONQW subsystem parameter specifies the maximum length of time
that a client connection waits for a DBAT to process the next unit-of-work or new
connection request.

Acceptable values: OFF, ON, 5 - 3600 seconds
Default: OFF
Update: option 46 on panel DSNTIPB
DSNZPxxx: DSN6FAC MAXCONQW

ON Connections wait as long as the value specified by the IDTHTOIN subsystem parameter.

OFF Connections wait indefinitely for a DBAT to process requests.

Numeric value
The time duration in seconds that a connection waits for a DBAT to process the request.

Each queued connection request entry is examined to see if its time waiting in the queue has exceeded the specified value. If the time is exceeded, the client connection is closed. The function is enabled only when Db2 subsystem is a member of a data sharing group.

This subsystem parameter can be updated online.

Related reference:
IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter)

DSNTIPX: Routine parameters panel
The entries on this panel are used to start the stored procedures address space so that it can run stored procedures or user-defined functions.

Figure 51. Routine parameters panel: DSNTIPX

Related concepts:
Migration step 23: Configure Db2 for running stored procedures and user-defined functions (optional)
WLM management of stored procedures
WLM PROC NAME field

The value of the WLM PROC NAME field specifies a name for the stored procedures JCL procedure that is to be generated during installation. This JCL procedure is generated for the default WLM environment that is specified in the WLM ENVIRONMENT field.

Acceptable values: 1 - 8 alphanumeric characters
Default: ssnmWLM
DSNZPxxx: none

This procedure is used for a WLM-established stored procedures address space.

If this field has a blank, the JCL procedure is still generated. In this case, the JCL procedure will be named by appending the string WLM to the Db2 subsystem name (specified on panel DSNTIPM in the field SUBSYSTEM NAME).

NUMBER OF TCBS field

The NUMBER OF TCBS field determines the number of SQL CALL statements and invocations of user-defined functions that can be processed concurrently in the address space of the default WLM environment. The default WLM environment is specified in the WLM ENVIRONMENT field.

Acceptable values: 1 - 100
Default: 8
DSNZPxxx: none

The larger the value, the more stored procedures and user-defined functions you can run concurrently in one address space. This value is dependent on the z/OS UNIX System Services MAXPROCUSER value. If this value is set above the z/OS UNIX System Services MAXPROCUSER value, you may exceed the maximum number of processes for the user.

Related tasks:

Maximizing the number of procedures or functions that run in an address space (Db2 Performance)

Related reference:

WLM ENVIRONMENT field (WLMENV subsystem parameter)

MAX ABEND COUNT field (STORMXAB subsystem parameter)

The STORMXAB subsystem parameter specifies the number of times a stored procedure or an invocation of a user-defined function is to be allowed to terminate abnormally. If this threshold is reached, SQL CALL statements for the stored procedure or user-defined function are rejected.

Acceptable values: 0 - 255
Default: 0
Update: option 47 on panel DSNTIPB
DSNZPxxx: DSN6SYSP STORMXAB

The default of 0 means that the first abend of a stored procedure or user defined function causes SQL CALL statements to that procedure or function to be rejected. For production systems, you should accept the default.
**TIMEOUT VALUE field (STORTIME subsystem parameter)**
The STORTIME subsystem parameter specifies the number of seconds before Db2 is to stop waiting for an SQL CALL statement or invocation of a user-defined function. This parameter applies to user-defined functions that are to be assigned to one of the task control blocks (TCBs) in a Db2 stored procedures address space.

Acceptable values: 5 - 1800, NOLIMIT  
Default: 180  
Update: option 47 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP STORTIME

If the time interval expires, the SQL statement fails. The default is a reasonable waiting time for most sites. You might want to choose a higher value if your system has long queues. You might want to choose a lower value if you want to minimize the waiting time for end-user requests. The NOLIMIT value means that Db2 waits indefinitely for the SQL request to complete, while the thread is active.

**Recommendation:** Do not select the NOLIMIT value. If the stored procedure address space is down for some reason or the user-defined function does not complete, your SQL request hangs until the request is satisfied or the thread is canceled.

**WLM ENVIRONMENT field (WLMENV subsystem parameter)**
The WLMENV subsystem parameter specifies the name of the default WLM_ENVIRONMENT that is to be used for a user-defined function or stored procedure. This environment is used when a value is not specified for the WLM_ENVIRONMENT option on the CREATE FUNCTION or CREATE PROCEDURE statements.

Acceptable values: Any valid name from 1 - 32 alphanumeric characters, including underscores  
Default: blank  
Update: option 47 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP WLMENV

Changing this value does not change existing routines because the value is stored in the catalog when the function or procedure is created.

**MAX OPEN CURSORS field (MAX_NUM_CUR subsystem parameter)**
The MAX_NUM_CUR subsystem parameter specifies the maximum number of cursors, including allocated cursors, that can be open at a given Db2 site for each thread.

Acceptable values: 0 - 99999  
Default: 500  
Update: option 47 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM MAX_NUM_CUR

Db2 keeps a count of currently open cursors. If an application attempts to open a thread after the maximum is reached, the statement will fail.
MAX STORED PROCS field (MAX_ST_PROC subsystem parameter)
The MAX_ST_PROC subsystem parameter determines the maximum number of stored procedures that are to be allowed for each thread.

Acceptable values: 0 - 99999
Default: 2000
Update: option 47 on panel DSNTIPB
DSNZPxxx: DSN6SPRM MAX_ST_PROC

If an application attempts to call a stored procedure after the maximum is reached, the statement will fail.

BIF COMPATIBILITY field (BIF_COMPATIBILITY subsystem parameter)
The BIF_COMPATIBILITY subsystem parameter specifies whether built-in functions and specifications are to return results in the newer format or revert to the pre-DB2 10 format.

Recommendation: Avoid making online changes to the BIF_COMPATIBILITY setting, even though they are supported. Such changes do not affect SQL statements in the dynamic SQL statement cache.

Acceptable values: V9_DECIMALVARCHAR, V9_TRIM, CURRENT, or V9
Default: For installation: CURRENT
For migration: the setting that your Db2 11 DSNTIDX or DSNTIDX input member specifies
Update: option 47 on panel DSNTIPB
DSNZPxxx: DSN6SPRM.BIF_COMPATIBILITY

In Db2 12, when the input data to the CHAR or VARCHAR function is a decimal value or a decimal is input to a CAST specification where the target data type is CHAR or VARCHAR, the result does not have any of the following attributes:

- Leading zeros.
- A trailing decimal point character.
- Leading blanks for positive decimal values.

You might need to modify some of your applications to handle this change, which differs from the behavior in DB2 9. While you prepare for this change, you can use the following options of the BIF_COMPATIBILITY subsystem parameter to temporarily revert to the DB2 9 behavior.

CURRENT
All built-in functions return data in the Db2 12 format, unless they are controlled by the APPLCOMPAT bind option. When you run the installation CLIST in INSTALL mode, CURRENT is the default setting.

V9_DECIMALVARCHAR
For decimal input, the CHAR and VARCHAR functions, and the CAST specification with decimal input and a target data type of CHAR or VARCHAR, return data in the DB2 9 format.
Tip: You can run with BIF_COMPATIBILITY set to CURRENT, and use the CHAR9 or VARCHAR9 functions instead for applications that depend on the DB2 9 behavior for the CHAR and VARCHAR functions with decimal input.

Db2 also continues to allow some unsupported timestamp strings for applications. Furthermore, as in previous Db2 releases, SQLCODE -180 is not issued for these unsupported timestamp strings. Because these strings are not supported by Db2, you must modify the application SQL to use a supported format for a timestamp string. For the supported formats, see String representations of datetime values (Db2 SQL).

V9_TRIM

The V9_DECIMAL VARCHAR setting rules are in effect, and if the APPLCOMPAT bind option is V10R1, the LTRIM, RTRIM and STRIP built-in functions behave like in DB2 9 with respect to tolerating invalid EBCDIC mixed data strings and treatment of EBCDIC control characters X’0E0F’.

V9

For decimal input, the CHAR built-in function returns data in the DB2 9 format.

This value has no effect on the output of the VARCHAR built-in function with a decimal value as input or on the result of the CAST(decimal-expression AS VARCHAR) specification. Unless you need DB2 9 behavior only for CHAR output, specify the V9_DECIMAL VARCHAR value.

Examples

The following examples show the effect of the BIF_COMPATIBILITY subsystem parameter on the output of these built-in functions and cast specifications.

Table 38. Example output for CHAR based on BIF_COMPATIBILITY settings

<table>
<thead>
<tr>
<th>BIF_COMPATIBILITY setting</th>
<th>OUTPUT for CHAR(000.1)</th>
<th>OUTPUT for CHAR(1000.)</th>
<th>OUTPUT for CHAR(1.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V9_DECIMAL VARCHAR</td>
<td>'000.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
<tr>
<td>CURRENT</td>
<td>'.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
<tr>
<td>V9</td>
<td>'000.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
</tbody>
</table>

Table 39. Example output for VARCHAR and CAST(decimal-expression AS VARCHAR) based on BIF_COMPATIBILITY settings

<table>
<thead>
<tr>
<th>BIF_COMPATIBILITY setting</th>
<th>Output for VARCHAR(000.1) or CAST(000.1 AS VARCHAR(3))</th>
<th>Output for VARCHAR(1000.) or CAST(1000. AS VARCHAR(5))</th>
<th>Output for VARCHAR(1.1) or CAST(1.1 AS VARCHAR(3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>V9_DECIMAL VARCHAR</td>
<td>'0.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
<tr>
<td>CURRENT</td>
<td>'.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
</tbody>
</table>

Table 40. Example output for CAST(decimal-expression AS CHAR) based on BIF_COMPATIBILITY settings

<table>
<thead>
<tr>
<th>BIF_COMPATIBILITY setting</th>
<th>Output for CAST(000.1 AS CHAR(6))</th>
<th>Output for CAST(1000. AS CHAR(6))</th>
<th>Output for CAST(1.1 AS CHAR(4))</th>
</tr>
</thead>
<tbody>
<tr>
<td>V9_DECIMAL VARCHAR</td>
<td>'.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
<tr>
<td>CURRENT</td>
<td>'.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
<tr>
<td>V9</td>
<td>'.1'</td>
<td>'1000.'</td>
<td>'1.1'</td>
</tr>
</tbody>
</table>
Related reference:
- CHAR (Db2 SQL)
- CHAR9 (Db2 SQL)
- VARCHAR (Db2 SQL)
- VARCHAR9 (Db2 SQL)

**ADMIN SCHEDULER field (ADMTPROC subsystem parameter)**
The ADMIN SCHEDULER field identifies a name for the JCL procedure that is used to start the Db2 administrative task scheduler that is associated with this Db2 subsystem.

Acceptable values: 1 - 8 alphanumeric characters
Default: ssnADMT
Update: option 47 on panel DSNTIPB
DSNZPxxx: DSN6SPRM ADMTPROC

### 1 - 8 alphanumeric characters
When Db2 starts, it starts the administrative task scheduler that is specified in the ADMIN SCHEDULER field, if that scheduler is not yet started. Also, every time Db2 starts or stops, it posts an event to the administrative task scheduler so that the administrative task scheduler can execute tasks that depend on those events.

**blank**
To disable the administrative task scheduler, enter blanks in the ADMIN SCHEDULER field. If you replace the default value with blanks, Db2 will not start the administrative task scheduler until you update the subsystem parameter.

This parameter cannot be updated online.

**DSNTIPX1: Db2 Java properties panel**
The entries on this panel are used to specify local preferences for configuring JDBC and SQLJ and DSNWLMJ_JAVA.

The fields on this panel can be updated only when you run the DB2 CLIST to install or migrate a Db2 subsystem or the first member of a data sharing group. They are displayed but cannot be changed when you install or migrate the second or a subsequent member of a data sharing group.
**z/OS SDK HOME DIR field**

The absolute path of the HFS/zFS directory where the z/OS Java Software Development Kit (SDK) resides. If you do not use Java, just accept the default setting.

- **Acceptable values:** An HFS/zFS path name of up to 47 characters
- **Default:** /usr/lpp/java150/J5.0
- **Update:** None
- **Data sharing scope:** Group
- **DSNZPxxx:** None
- **z/OSMF variable:** JAVAHOME

This directory is provided by z/OS and is referenced by the JAVA_HOME variable in the file specified in the JAVA ENV VAR FILE field.

Note that the 64-bit JVM environment requires use of Java 6 Software Development Kit (SDK) or higher.

**Z/OS USS JAVA CLASSES field**

The absolute path of the HFS/zFS directory that contains JAR files provided by z/OS UNIX System Services (z/OS UNIX).

Verify that this directory contains the following JAR files:
- `vgxljapi.jar`: z/OS XML System Services XML schema registration
- `IRRacf.jar`: The Java interface to the system security product

- **Acceptable values:** An HFS/zFS path name of up to 47 characters
- **Default:** /usr/include/java_classes/
- **Update:** None
- **Data sharing scope:** Group
- **DSNZPxxx:** None
- **z/OSMF variable:** CLASPATH
This directory is provided by z/OS and is referenced by the CLASSPATH variable in the file that is specified in the JAVA ENV VAR FILE field. If you do not use Java, accept the default setting.

**z/OS USS JAVA RUNTIME field**

The absolute path of the HFS/zFS directory that contains the native library (DLLs) provided by z/OS Unix System Services (z/OS Unix). Verify that this directory contains libGXLNVBAT.so, the DLL for z/OS XML System Services.

Acceptable values: An HFS/zFS path name of up to 47 characters
Default: /usr/lib/java_runtime
Update: None
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: LIBPATH

This directory is provided by z/OS and is referenced by the LIBPATH variable in the file specified in the JAVA ENV VAR FILE field. If you do not use Java, just accept the default setting.

**JDBC/SQLJ ENV VAR DIR field**

The absolute path of the highest-level directory in the set of HFS/zFS directories that contain the classes and other objects for the IBM Data Server Driver for JDBC and SQLJ on Db2 12 for z/OS.

Acceptable values: An HFS/zFS path name of up to 47 characters
Default: /usr/lpp/db2c10/jdbc
Update: None
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: JCCHOME

This directory and its subdirectories are typically managed by SMP/E. It is referenced by the JCC_HOME variable in the file specified in the JAVA ENV VAR FILE field. Verify that the directory specified here contains the following subdirectories which are used for Db2 configuration:

- bin: Contains the runtime objects for the IBM Data Server Driver for JDBC and SQLJ. This subdirectory is referenced by the JDBC and SQLJ PATH variable.
- classes: Contains the JAR files for the IBM Data Server Driver for JDBC and SQLJ. This subdirectory is referenced by the JDBC and SQLJ CLASSPATH variable.
- lib: Contains the DLLs for the IBM Data Server Driver for JDBC and SQLJ. This subdirectory is referenced by the JDBC and SQLJ LIBPATH variable.

This directory set exists only if you have received and applied JDBCC12, the FMID for IBM Data Server Driver for JDBC and SQLJ on Db2 12 for z/OS. If you have not, just accept the default setting.

**JDBC/SQLJ INSTALL ID field**

The ID for connecting to Db2 using a secure sign on and a RACF PassTicket for authentication. The ID will be used for binding the IBM Data Server Driver for JDBC and SQLJ during installation or migration to Db2 and for running an installation verification procedure (IVP) to validate that JDBC and SQLJ are deployed correctly on Db2.
Acceptable values: A user ID of 1 to 8 characters
Default: The value of SECURITY ADMIN 1 on panel DSNTIPPI
Update: None
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: JCCSID

This ID is meaningful only if you have received and applied JDBCC12, the FMID for Db2 12 JDBC/SQLJ. If you have not, just accept the default setting.

**DB2 BASE JAVA DIR field**
The absolute path of the highest-level directory in the set of HFS/zFS directories that contain the Db2 for z/OS-supplied Java classes, native methods (DLLs), and samples.

Acceptable values: An HFS/zFS path name of up to 47 characters
Default: /usr/lpp/db2c10/base
Update: None
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: DB2BASE

This directory and its subdirectories are typically managed by SMP/E. It is referenced by the DB2_BASE variable in the file specified in the JAVA ENV VAR FILE field. Verify that the directory specified here contains the following subdirectories which are used for Db2 configuration:

- classes: Contains the DB2 base-supplied jar files.
- lib: Contains the DB2 base-supplied DLLs.
- samples: Contains the DB2 base-supplied sample objects.

If you do not use Java, just accept the default setting.

**JAVA ENV DATA SET field**
The name of the environment (JAVAENV) data set for the Db2-supplied Java WLM Java Environment, DSNWLM_JAVA.

Acceptable values: An data set name of up to 44 characters
Default: prefix.ssnm WLMJJAVAENV or prefix.group-attach-name WLMJJAVAENV (if a group attach name was specified on panel DSNTIPK)
Update: None
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: JAVAENV

This data set is created by job DSNTIJMJ and must not already exist. It is allocated by the JAVAENV DD of DSNWLMJ, the address space startup procedure for DSNWLM_JAVA. DSNWLMJ is created by job DSNTIJMW. It contains the _CEE_ENVFILE environment variable which provides the name and location of the Java environment variable file to be used when the DSNWLM_JAVA environment is started.
The security profile of the user ID used to execute job DSNTIJMJ must include an OMVS segment. The user ID must also be authorized to create the specified data set.

If you do not use Java, just accept the default setting.

**JAVA ENV VAR FILE field**
The absolute path and name of the HFS/zFS environment variable file for the Db2-supplied Java WLM environment, DSNWLM_JAVA.

- **Acceptable values:** An HFS/zFS path name of up to 47 characters
- **Default:** /usr/lpp/db2c10/base/classes/ssnmenvfile.txt or /usr/lpp/db2c10/base/classes/group-attach-nameenvfile.txt (if a group attach name was specified on panel DSNTIPK)
- **Update:** None
- **Data sharing scope:** Group
- **DSNZPxxx:** None
- **z/OSMF variable:** JAVAENVV

This file is created when you run installation job DSNTIJMJ and must not already exist. It is referenced in the _CEE_ENVFILE environment variable of the JAVAENV data set specified in the JAVA ENV DATA SET field. It specifies the names and locations of the following properties to be used when the DSNWLM_JAVA environment is started:

- **CLASSPATH:** The HFS/zFS directory of the Java class that contains the jar file for z/OS XML System Services XML schema registration (gxljapi.jar).
- **DB2_BASE:** The highest-level directory in the set of HFS/zFS directories that contain DB2 for z/OS-supplied Java classes and native methods (DLLs).
- **JCC_HOME:** The highest-level directory in the set of HFS/zFS directories that contain the classes and other objects for the IBM Data Server Driver for JDBC and SQLJ.
- **JAVA_HOME:** The HFS/zFS directory where the z/OS Java Software Development Kit (SDK) resides.
- **JVMPROPS:** The HFS/zFS path and name of the startup options for the Java Virtual Machine (JVM) used by the DB2-supplied Java WLM environment, DSNWLM_JAVA.
- **LIBPATH:** The HFS/zFS directory that contains the native library for z/OS XML System Services (libGXLNVBAT.so).
- **STEPLIB:** The name of the DB2 for z/OS runtime library.

The security profile of the user ID used to execute job DSNTIJMJ must include an OMVS segment. The user ID must also have write permission to the HFS directory where the file is to be created.

If you do not use Java, just accept the default setting.

**JVM PROPERTIES FILE field**
The absolute path and name of the HFS/zFS Java Virtual Machine (JVM) startup options file used by the Db2-supplied Java WLM environment, DSNWLM_JAVA.

- **Acceptable values:** An HFS/zFS path name of up to 47 characters
- **Default:** /usr/lpp/java/properties/ssnmmvmsp or /usr/lpp/java/properties/group-attach-namevmsp (if a group attach name was specified on panel DSNTIPK)
This file is created when you run installation job DSNTIJMJ and must not already exist. It is referenced by the JVMPROPS variable in the file specified in the JAVA ENV VAR FILE field.

The security profile of the user ID used to execute job DSNTIJMJ must include an OMVS segment. The user ID must also have write permission to the HFS directory where the file is to be created.

If you do not use Java, just accept the default setting.

**JVM ADDRESSING MODE field**
Indicates whether DSNWLMJ, the address space proc for DSNWLM_JAVA, should be configured to request a 32-bit or a 64-bit JVM.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>32 or 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>32</td>
</tr>
<tr>
<td>Update:</td>
<td>None</td>
</tr>
<tr>
<td>Data sharing scope:</td>
<td>Group</td>
</tr>
<tr>
<td>DSNZPxxx:</td>
<td>None</td>
</tr>
<tr>
<td>z/OSMF variable:</td>
<td>JVMAMODE</td>
</tr>
</tbody>
</table>

The DSNWLMJ address space startup proc is added to the system proclib by installation job DSNTIJMW.

When this field is set to 32, DSNTIJMW is configured to add DSNWLMJ with the following characteristics:
- The NUMTCB setting is 5
- The started module is called DSNX9WLM

When this field is set to 64, DSNTIJMW is configured to add DSNWLMJ has the following characteristics:
- The NUMTCB setting is 25
- The started module is called DSNX9WJM

Note that the 64-bit JVM environment requires use of Java 6 Software Development Kit (SDK) or higher. The SDK is specified in field Z/OS SDK HOME DIR.

If you do not use Java, just accept the default setting.

**INIT JAVA HEAP SIZE field**
Indicates the size in megabytes of the initial Java heap (the middleware heap within non-system heap) for the JVM used by the DSNWLM_JAVA WLM environment. This value is assigned to the -Xms option in the JVM startup options file specified in the JVM PROPERTIES FILE field.

<table>
<thead>
<tr>
<th>Acceptable values:</th>
<th>DEFAULT or 1 - 99999 (in megabytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default:</td>
<td>32</td>
</tr>
<tr>
<td>Update:</td>
<td>None</td>
</tr>
</tbody>
</table>
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: JVMHEAP

If you specify DEFAULT, the -Xms option is assigned 64M, but remains commented out in the JVM startup options file. If you specify a numeric value, it is assigned to the -Xms option and the option is active in the JVM startup options file. Note that the CLIST has no way to validate whether system resources are available to support the specified size.

If you do not use Java, just accept the default setting.

**MAX APPL HEAP SIZE field**
Indicates the maximum size in megabytes of memory for application work (non-system heap) in the JVM used by the DSNWLM_JAVA WLM environment. This value is assigned to the -Xmx option in the JVM startup options file specified in the JVM PROPERTIES FILE field.

Acceptable values:: DEFAULT or 1 - 99999 (in megabytes)
Default: DEFAULT
Update: None
Data sharing scope: Group
DSNZPxxx: None
z/OSMF variable: JVMAPPH

If you specify DEFAULT, the -Xmx option is assigned 128M but remains commented out in the JVM startup options file. If you specify a numeric value, it is assigned to the -Xmx option and the option is active (uncommented) in the JVM startup options file. Note that the CLIST has no way to validate whether system resources are available to support the specified size.

If you do not use Java, just accept the default setting.

**DSNTPR1: Parameters for Db2-supplied routines panel**
The DSNTPR1 panel enables you to establish default WLM environment names for Db2-supplied routines and to access panels for advanced configuration of these routines.
Values on the DSNTIPR1 panel are used to customize configuration statements in installation job DSNTIJRT, which creates and grants access to the Db2-supplied routines and any related databases, table spaces, tables, indexes, and created global temporary tables. The values are also used to customize installation job DSNTIJRW, which defines core WLM environments for Db2-supplied stored procedures and user-defined functions. If you change an environment name on the DSNTIPR1 panel, the change is applied in the advanced configuration panel settings to all routines that use the original name.

If you change a WLM environment name on this panel, the WLM ENV NAME field of each corresponding routine in the advanced configuration options panels is also changed. For example, if you change the setting of the FOR DB2 XML PROCESSING ROUTINES field from DSNWLM_XML to DSNWLM_MY_XML, then the DSNTIPRJ panel will contain DSNWLM_MY_XML in the WLM ENV NAME fields for all routines.

Changes on the DSNTIPR1 panel do not affect a corresponding routine's WLM ENV NAME field if the WLM ENV NAME value has been previously modified on the advanced configuration options panel. For example, if you change the WLM ENV NAME setting for XSR_ADDSCHEMADOC from DSNWLM_XML to MY_WLM_ENVIRONMENT on the DSNTIPRJ panel, it will remain MY_WLM_ENVIRONMENT even if you subsequently change the setting of the FOR DB2 XML PROCESSING ROUTINES field on the DSNTIPR1 panel from DSNWLM_XML to DSNWLM_MY_XML.

**FOR MOST Db2 ROUTINES field**
The value of the FOR MOST Db2 ROUTINES field identifies the WLM environment that is to be used for running most Db2-supplied routines.

Acceptable values: a valid WLM environment name
Default: DSNWLM_GENERAL
DSNZPxxx: none

**Related reference:**
Installation information for procedures and functions that are supplied with Db2
FOR ROUTINES IDENTIFIED TO PROGRAM CONTROL field
The FOR ROUTINES IDENTIFIED TO PROGRAM CONTROL field identifies the WLM environment that is to be used for running Db2-supplied routines that require program control. These routines can be registered to RACF or an equivalent product.

Acceptable values: a valid WLM environment name
Default: DSNWLM_PGM_CONTROL
DSNZPxxx: none

Related reference:
Installation information for procedures and functions that are supplied with Db2
Related information:
Defining modules to program control (UNIX System Services Planning)

FOR SERIALIZED ROUTINES field
The value of the FOR SERIALIZED ROUTINES field identifies the WLM environment that is to be used for Db2-supplied routines that need to run serially. A routine that must run serially cannot share a task control block with another routine.

These routines require an environment that has a NUMTCB setting of 1. These routines do not include LANGUAGE REXX routines or SYSPROC.DSNUTILS and SYSPROC.DSNUTILU.

Acceptable values: a valid WLM environment name
Default: DSNWLM_NUMTCB1
DSNZPxxx: none

FOR Db2 UTILITIES ROUTINES field
The value of the FOR Db2 UTILITIES ROUTINES field identifies the name of the WLM environment that is to be used for running the Db2-supplied routines for utilities. This WLM environment should be used only by the SYSPROC.DSNUTILS and SYSPROC.DSNUTILU stored procedures.

Acceptable values: a valid WLM environment name
Default: DSNWLM_UTILS
DSNZPxxx: none

Related reference:
Installation information for procedures and functions that are supplied with Db2

FOR Db2 XML PROCESSING ROUTINES field
The value of the FOR Db2 XML PROCESSING ROUTINES field identifies the WLM environment that is to be used for running most Db2-supplied XML processing routines. This applies to the XSR_ADDSCHEMADOC, XSR_REGISTER, and XSR_REMOVE routines.

Acceptable values: a valid WLM environment name
Default: DSNWLM_XML
DSNZPxxx: none

Related reference:
Installation information for procedures and functions that are supplied with Db2
**FOR MOST Db2 JAVA ROUTINES field**
The value of the FOR MOST Db2 JAVA ROUTINES field identifies the WLM environment that is to be used for running most Db2-supplied Java routines.

Acceptable values: a valid WLM environment name
Default: DSNWLM_JAVA
DSNZPxxx: none

Related reference: Installation information for procedures and functions that are supplied with Db2

**FOR Db2 REXX ROUTINES field**
The value of the FOR Db2 REXX ROUTINES field identifies the name of the WLM environment that is to be used for Db2-supplied REXX routines.

Acceptable values: a valid WLM environment name
Default: DSNWLM_REXX
DSNZPxxx: none

Related reference: Installation information for procedures and functions that are supplied with Db2

**FOR UNIFIED DEBUGGER ROUTINES field**
The FOR UNIFIED DEBUGGER ROUTINES field identifies the WLM environment that is to be used for Db2-supplied routines that are for the unified debugger tool.

Acceptable values: a valid WLM environment name
Default: DSNWLM_DEBUGGER
DSNZPxxx: none

Related reference: Installation information for procedures and functions that are supplied with Db2

**FOR THE Db2 CICS ROUTINE field**
The value of the FOR THE Db2 CICS ROUTINE field identifies the WLM environment that is to be used for the Db2 CICS transaction processor routine SYSPROC.DSNACICS.

Acceptable values: a valid WLM environment name
Default: DSNWLM_CICS
DSNZPxxx: none

Related reference: Installation information for procedures and functions that are supplied with Db2

**FOR Db2 MQSERIES ROUTINES field**
The value of the FOR Db2 MQSERIES ROUTINES field specifies the name of the WLM environment that is to be used for Db2 MQSeries® functions.

Acceptable values: a valid WLM environment name
Default: DSNWLM_MQSERIES
DSNZPxxx: none

Related reference: Installation information for procedures and functions that are supplied with Db2
FOR Db2 WEB SERVICES ROUTINES field

The value of the FOR Db2 WEB SERVICES ROUTINES field identifies the WLM environment that is to be used for Db2 web services functions.

Acceptable values: a valid WLM environment name
Default: DSNWLM_WEB SERVICES
DSNZPxxx: none

Related reference:
[Installation information for procedures and functions that are supplied with Db2]

ADVANCED CONFIGURATION OPTIONS field

The value of the ADVANCED CONFIGURATION OPTIONS field determines whether optional panels that are for advanced configuration of Db2-supplied stored procedures and functions are to be displayed.

Acceptable values: NO, YES
Default: NO
DSNZPxxx: none

NO  Skips the panels for advanced configuration of stored procedures and displays the DSNTIPZ panel.

YES  Displays panels for advanced configuration of Db2-supplied stored procedures, starting with panel DSNTIPRA.

Db2 routine selection panel: DSNTIPRA

The DSNTIPRA panel provides access to additional panels for advanced configuration options of Db2-supplied routines. This panel is displayed if you specify YES in the ADVANCED CONFIGURATION OPTIONS field on the DSNTIPRI panel.

If you specify NO in the ADVANCED CONFIGURATION OPTIONS field, the DSNTIPZ panel is displayed instead.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
To view and modify the advanced options for a routine type:
In the Option field, enter the number of the routine type, and then press the
Enter key.

To exit advanced configuration and display the next standard installation
panel, DSNTIPZ:
Leave the Option field blank and press the Enter key.

To return to the DSNTIPRA panel:
Press the Return key.

Trace formatting routines panel: DSNTIPRB:

The DSNTIPRB panel allows you to modify configuration options for trace
formatting routines.

Because space on this panel is limited, all fields might not be displayed at one
time. The number of fields that are displayed depends on the maximum number of
rows your ISPF display device can display at one time, whether you use split
screen mode, and where you choose to split the screen. This panel is defined as a
scrollable panel, which enables you to view all the fields if they are not displayed
at one time. The panel can be scrolled when the label "More" displays near the
upper right corner of the panel. A plus symbol (+) following "More" indicates that
the panel can be scrolled down, and a minus symbol (-) indicates that the panel
can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll
through the fields. (Alternatively, you can assign your PF keys to be the DOWN
and UP commands.)
WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_NUMTCB1
DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

Utility invocation routines panel: DSNTIPRC:

The DSNTIPRC panel allows you to modify configuration options for utility invocation routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll...
through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)

WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_UTILS
DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

Db2 security routines panel: DSNTIPRE:

The DSNTIPRE panel allows you to modify configuration options for the SYSIBM.USERNAMES encryption routine.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split
screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)

Figure 57. Db2 security routines panel: DSNTIPRE

**WLM ENV NAME**

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_NUMTCB1
DSNZPxxx: none

**GRANT EXECUTE**

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

**PACKAGE OWNER**

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

**Java and SQL procedure processing routines panel: DSNTIPRF:**

The DSNTIPRF panel allows you to modify configuration options for SQL procedure processing routines.
Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)

<p>| +---------- DB2 ROUTINES: ADVANCED CONFIGURATION OPTIONS +---------- |</p>
<table>
<thead>
<tr>
<th>DSNTPRF  Java and SQL procedure processing routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER_JAVA_PATH    ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>DB2_INSTALL_JAR   ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>DB2_REMOVE_JAR    ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>DB2_REPLACE_JAR   ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>DB2_UPDATEJARINFO ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>INSTALL_JAR       ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>REMOVE_JAR        ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>REPLACE_JAR       ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>DSNTBIND          ===&gt; DSNWLM_REXX</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_REXX</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
<tr>
<td>DSNTPSMP          ===&gt; DSNWLM_REXX</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_REXX</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>WLM_REFRESH       ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>WLM_ENV_NAME      ===&gt; DSNWLM_GENERAL</td>
</tr>
<tr>
<td>GRANT EXECUTE    ===&gt; PUBLIC</td>
</tr>
<tr>
<td>PACKAGE OWNER    ===&gt;</td>
</tr>
</tbody>
</table>

PRESS: ENTER to continue UP/DOWN to scroll RETURN to exit

Figure 58. Java and SQL procedure processing routines panel: DSNTPRF

**WLM ENV NAME**

Specify the WLM environment name for the routine.
Acceptable values: a valid WLM environment name
Default: DSNWLM_GENERAL for most routines, DSNWLM_REXX for DSNTBIND and DSNTPSMP
DSNZPxxx: none

**GRANT EXECUTE**

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

**PACKAGE OWNER**

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

**Unified debugger for Java and SQL routines panel: DSNTIPRH:**

The DSNTIPRH panel allows you to modify configuration options for unified debugger for SQL and Java routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLMD_DEBUGGER for most routines,
DSNWLM_GENERAL for DBG_RUNSESSIONMANAGER

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.
Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZP: none

**PACKAGE OWNER**

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZP: none

**XML processing routines panel: DSNTIPRI:**

The DSNTIPRI panel allows you to modify configuration options for XSL transform and XML schema processing routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_XML for most routines, DSNWLM_JAVA for XSR_COMPLETE and XSLTRANSFORM

DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC

DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank

DSNZPxxx: none
IBM Text Search routines panel: DSNTIPRJ:

The DSNTIPRJ panel allows you to modify configuration options for Text Search routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)

Figure 61. IBM Text Search routines panel: DSNTIPRJ

WLM ENV NAME

Specify the WLM environment name for the routine.
Acceptable values: a valid WLM environment name
Default: DSNWLM_GENERAL (DSNWLM_JAVA for SYSTS_ENCRYPT)
DSNZPxxx: none

**GRANT EXECUTE**

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

**PACKAGE OWNER**

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

**Database metadata routines panel: DSNTIPRK:**

The DSNTIPRK panel allows you to modify configuration options for database metadata routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
Figure 62. Database metadata routines panel: DSNTIPRK
**WLM ENV NAME**

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_GENERAL
DSNZPxxx: none

**GRANT EXECUTE**

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

**PACKAGE OWNER**

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

**Db2 administration routines panel: DSNTPRL:**

The DSNTPRL panel allows you to modify configuration options for routines that are to be used for administration.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
<table>
<thead>
<tr>
<th>Command</th>
<th>WLM ENV NAME</th>
<th>Grant Execute</th>
<th>Package Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIN_COMMAND_DB2</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_COMMAND_DSN</td>
<td>DSNWLMREXX</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_COMMAND_MVS</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_COMMAND_UNIX</td>
<td>DSNWLPGM_CONTROL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_BROWSE</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_DELETE</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_LIST</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_RENAME</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_SEARCH</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_WRITE</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_EXPLAIN_MAINT</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_INFO_HOST</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_INFO_SMS</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_INFO_SQL</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
<tr>
<td>ADMIN_INFO_SSID</td>
<td>DSNWLMGENERAL</td>
<td>PUBLIC</td>
<td></td>
</tr>
</tbody>
</table>

Figure 63. DB2 administration routines panel: DSNTIPRL (part 1 of 3)
Figure 64. DB2 administration routines panel: DSNTIPRL (part 2 of 3)
WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: see the panel illustration for the default environment name of each routine
DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for
most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

Db2 optimization routines panel: DSNTIPRM:

The DSNTIPRM panel allows you to modify configuration options for routines that are to be used for optimization.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)

WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_GENERAL
DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

The setting user-id is the TSO user ID used to run the installation CLIST.

Acceptable values: at least one valid authorization ID
Default: user-id
DSNZPxxx: none

PACKAGE OWNER

Figure 66. Db2 query optimization routines panel: DSNTIPRM
Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the
BIND PACKAGE command. This field does not display for routines that do not have package.

The setting user-id is the TSO user ID used to run the installation CLIST.

Acceptable values: blank, or a valid authorization ID  
Default: user-id  
DSNZPxxx: none

CICS and IMS transaction invocation routines panel: DSNTIPRN:

The DSNTIPRN panel allows you to modify configuration options for CICS and IMS transaction invocation routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label “More” displays near the upper right corner of the panel. A plus symbol (+) following “More” indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternately, you can assign your PF keys to be the DOWN and UP commands.)

Figure 67. CICS and IMS transaction invocation routines panel: DSNTIPRN

WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name  
Default: DSNWLM_GENERAL (DSNWLM_DSNACICS for DSNACICS)  
DSNZPxxx: none

GRANT EXECUTE
Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

MQSeries routines panel: DSNTIPRO:

The DSNTIPRO panel allows you to modify configuration options for MQSeries routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLMMQSERIES
DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for
most routines is blank, meaning that no OWNER parameter is included in the
BIND PACKAGE command. This field does not display for routines that do
not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

Web services/SOAP routines panel: DSNTIPRP:

The DSNTIPRP panel allows you to modify configuration options for web services
and SOAP routines.

Because space on this panel is limited, all fields might not be displayed at one
time. The number of fields that are displayed depends on the maximum number of
rows your ISPF display device can display at one time, whether you use split
screen mode, and where you choose to split the screen. This panel is defined as a
scrollable panel, which enables you to view all the fields if they are not displayed
at one time. The panel can be scrolled when the label "More" displays near the
upper right corner of the panel. A plus symbol (+) following "More" indicates that
the panel can be scrolled down, and a minus symbol (-) indicates that the panel
can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll
through the fields. (Alternatively, you can assign your PF keys to be the DOWN
and UP commands.)

Figure 69. Web services/SOAP routines panel: DSNTIPRP

WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values: a valid WLM environment name
Default: DSNWLM_WEBSERVICES
DSNZPxxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access
on the routine. Use commas to separate individual IDs. Do not use embedded
blanks. You can enter up to 44 characters, including commas.
Acceptable values: at least one valid authorization ID
Default: PUBLIC
DSNZPxxx: none

PACKAGE OWNER

Specify the ID that should own the package when it is bound (the value for the OWNER parameter of the BIND PACKAGE command). The default for most routines is blank, meaning that no OWNER parameter is included in the BIND PACKAGE command. This field does not display for routines that do not have package.

Acceptable values: blank, or a valid authorization ID
Default: blank
DSNZPxxx: none

JSON routines panel: DSNTIPRQ:

The DSNTIPRQ panel allows you to modify configuration options for JSON routines.

Because space on this panel is limited, all fields might not be displayed at one time. The number of fields that are displayed depends on the maximum number of rows your ISPF display device can display at one time, whether you use split screen mode, and where you choose to split the screen. This panel is defined as a scrollable panel, which enables you to view all the fields if they are not displayed at one time. The panel can be scrolled when the label "More" displays near the upper right corner of the panel. A plus symbol (+) following "More" indicates that the panel can be scrolled down, and a minus symbol (-) indicates that the panel can be scrolled up. You can use the ISPF scroll commands DOWN and UP to scroll through the fields. (Alternatively, you can assign your PF keys to be the DOWN and UP commands.)
WLM ENV NAME

Specify the WLM environment name for the routine.

Acceptable values:
- a valid WLM environment name
- DSNWLM_GENERAL
- DSNZP_xxx: none

GRANT EXECUTE

Specify one or more authorization IDs that are to be granted EXECUTE access on the routine. Use commas to separate individual IDs. Do not use embedded blanks. You can enter up to 44 characters, including commas.

Acceptable values:
- at least one valid authorization ID
- PUBLIC
- DSNZP_xxx: none

DSNTIPZ: Data definition control support panel

The entries on the DSNTIPZ panel allow you to install and tailor data definition control support.
Two SQL tables (application registration and object registration) are identified and created even if data definition control support is not installed. This simplifies future activation of the facility. Specified application identifiers (Db2 plans or collections of packages) can be registered in the application registration table, and, optionally, their associated Db2 object names can be registered in the object registration table. Db2 consults these two tables prior to accepting a given DDL statement to make sure that a particular application identifier and object name are registered.

**Related information:**

- Managing access through data definition control (Managing Security)

**INSTALL DD CONTROL SUPT field (RGFINSTL subsystem parameter)**

The RGFINSTL subsystem parameter determines whether data definition control support is to be installed.

```
Acceptable values: YES, NO
Default: NO
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFINSTL
```

**YES**

Data definition control support is installed.

**NO**

Data definition control support is not installed. DDL statements are not validated by this support. The application registration table and object registration table are still created according to values that are entered in fields 5 through 8 on panel DSNTIPZ.
CONTROL ALL APPLICATIONS field (RGFDEDPL subsystem parameter)
The RGFDEDPL subsystem parameter specifies whether the Db2 subsystem is to be completely controlled by a set of closed applications that are identified in the application registration table.

Acceptable values: YES, NO
Default: NO
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFDEDPL

Closed applications require their Db2 objects to be managed solely through the plans or packages of the closed application that is registered in the application registration table.

REQUIRE FULL NAMES field (RGFFULLQ subsystem parameter)
The RGFFULLQ subsystem parameter specifies whether registered objects are to require fully qualified names.

Acceptable values: YES, NO
Default: YES
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFFULLQ

UNREGISTERED DDL DEFAULT field (RGFDEFLT subsystem parameter)
The RGFDEFLT subsystem parameter specifies the action that is to be taken for DDL statements that name unregistered objects.

Acceptable values: ACCEPT, REJECT, APPL
Default: ACCEPT
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFDEFLT

ACCEPT
DDL statements that name unregistered objects are accepted.

REJECT
DDL statements that name unregistered objects are rejected.

APPL
DDL statements that name unregistered objects are rejected if the current application is not registered.

ART/ORT ESCAPE CHARACTER field (RGFESCP subsystem parameter)
The RGFESCP subsystem parameter specifies the escape character that is to be used in the application registration table (ART) or object registration table (ORT).

Acceptable values: any non-alphanumeric character
Default: none
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFESCP
Sets of names in the ART and ORT can be represented by patterns that use the underscore (_) and percent sign (%) characters in the same way as in an SQL LIKE predicate.

If you enter a character in this field, it can be used in those patterns in the same way as an escape character is used in an SQL LIKE predicate.

Related tasks:

- Installing data definition control support (Managing Security)

**REGISTRATION OWNER field (RGFCOLID subsystem parameter)**

The RGFCOLID subsystem parameter specifies the ID that is to own both the application registration table and the object registration table.

Acceptable values: 1 - 8 characters
Default: DSNRGCOL
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFCOLID

**REGISTRATION DATABASE field (RGFDBNAM subsystem parameter)**

The RGFDBNAM subsystem parameter specifies the name of the database that is to contain the registration tables that are used for data definition control support.

Acceptable values: 1 - 8 characters
Default: DSNRGFDB
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFDBNAM

**APPL REGISTRATION TABLE field (RGFNMPRT subsystem parameter)**

The RGFNMPRT subsystem parameter specifies the name that is to be used for the application registration table.

Acceptable values: 1 - 17 characters
Default: DSN_REGISTER_APPL
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFNMPRT

**OBJT REGISTRATION TABLE field (RGFNMORT subsystem parameter)**

The RGFNMORT subsystem parameter specifies the name that is to be used for the object registration table (ORT).

Acceptable values: 1 - 17 characters
Default: DSN_REGISTER_OBJT
Update: option 50 on panel DSNTIPB
DSNZPxxx: DSN6SPRM RGFNMORT

**DSNTIPY: Job editing panel**

The entries on the DSNTIPY panel specify values and information about job statements for the installation and sample application jobs.
The REMOTE LOCATION field specifies the location of another Db2 subsystem. The remote subsystem is used by the COBOL preparation sample job DSNTIJ3C, the DDF remote location update sample job DSNTIJ6, and the stored procedures sample jobs. The stored procedures sample jobs that require a remote location name are DSNTIJ6D, DSNTIJ6P, DSNTIJ6S, DSNTIJ6T, DSNTIJ61, and DSNTIJ62.

Acceptable values: 1 - 16 characters

Default: none

Update: See Updating subsystem parameter and application default values

DSNZPxxx: none

The location name must begin with a letter and must not contain special characters. A remote location name is accepted only if you have also entered a Db2 location name for the Db2 LOCATION NAME field on installation panel DSNTIPR.

Related reference:

**DB2 LOCATION NAME field**
Job card information fields
The installation job card information fields specify the job statements for all the installation jobs. The sample application job card information fields specify the job statements for all the sample application jobs.

Acceptable values: see below
Default: none
Update: option 51 on panel DSNTIPB
DSNZPxxx: none

Specify the job name in one of two ways:
• If the job name is member, the job name for each job is the same as its member name.
• If the job name is any value other than member, the name is truncated to seven characters, and one character is added to the end of the name to identify the run order for that job.

An example of job card information follows:
3====> //MEMBER JOB,
4====> // MSGLEVEL=(1,1), MSGCLASS=H, REGION=2M, CLASS=A
5====> // USER=SYSADM, PASSWORD=SYSADM, NOTIFY=SYSADM
6====>

If you use the installation SYSOPR authority for Db2 installation or migration, you must set the USER parameter in the job card for installation jobs to a primary or secondary authorization ID with the installation SYSOPR authority.

DSNTIPC: CLIST calculations panel 1
The DSNTIPC panel displays the messages that the installation CLIST issues to indicate calculated storage sizes.

The Override column of panel DSNTIPC controls whether Db2 uses the calculated value:
• To accept the calculated value, ensure that the Override column contains no value.
• To override the calculated value, specify a value in the Override column.

Space estimates from these messages do not account for cylinder rounding. Base requirements can be 10% - 20% higher than the message indicates, depending on the disk type.

The messages show that most of the needed virtual storage is in extended private storage and storage above the 2 GB bar, including the buffer pool, the EDM pool, and almost all of the code and working storage.

For most fields on this panel, you have a choice between accepting or overriding the calculated value:

During the tailoring session, message DSNT438I is issued to the tailoring terminal. This message, which shows the IRLM LOCK MAXIMUM SPACE and MAX STORAGE FOR LOCKS values, is always issued if you accept the default values.
This message indicates that the IRLM could request a total amount of space that is larger than the available space, causing an abend. The message is based on the following variables:

- The maximum number of data or row locks per user that is specified on installation panel DSNTIPJ (LOCKS PER USER)
- The number of users that is specified on installation panel DSNTIPE for MAX USERS and MAX REMOTE ACTIVE

The CLIST assumes that the private region that is available for IRLM locks is estimated as 60,000 KB.

### Related tasks:

- Calculating storage requirements with the installation CLIST
- Estimating the storage needed for locks (Db2 Performance)

### Related reference:

- DSNTIPD: Sizes panel 1

### Related information:

- DSNT438I (Db2 Messages)

#### DSMAX field (DSMAX subsystem parameter)

The DSMAX field specifies the maximum number of data sets that can be open at one time.

How Db2 enforces the limit depends on the DSMAX value:

- If the value is less than 32,768, when the DSMAX value is reached, new OPEN statements do not fail and are allowed to continue, unless the number of open data sets reaches 32,768.
- If the value is 32,768 or greater, when the DSMAX value is reached, new OPEN statements begin to fail with reason code 00C20113.
To minimize situations in which OPEN statements fail, Db2 tracks the number of open data sets. When the number of open data sets reaches the value of DSMAx, Db2 closes a number of data sets so that additional data sets can be opened.

Acceptable values: 1 - 200000
Default: based on calculations
Update: enter a value in the override column
DSNZPxxx: DSN6SPRM DSMAx

Although the maximum number of data sets is 200,000, the practical limit can be less than 30,000, depending on available storage below the line. The value that you enter can substantially influence the performance of Db2.

Complete one of the following actions to improve the performance of opening and closing data sets:
- Update the ALLOCxx parmlib member to set the SYSTEM MEMDSENQMGMT value to ENABLE.
- Issue system command SETALLOC SYSTEM,MEMDSENQMGMT=ENABLE.

It is best to update the ALLOCxx parmlib because the change remains effective across IPLs. If the SETALLOC command is used to enable SYSTEM MEMDSENQMGMT, you must restart Db2.

When a secondary index is nonpartitioned, the number of data sets that are required for the index is dependent on the total required space to contain the tree structure and on the size limit of each data set. When a secondary index is data-partitioned, the number of data sets that are required for the index is equal to the number of data partitions in the table space that contains the table. Unless a small piece size is used for nonpartitioned secondary indexes, partitioning generally results in an increase in the number of data sets for the index.

If the partitioning of secondary indexes causes the number of data sets to increase or decrease appreciably, you can modify the value of DSMAx. The default value for DSMAx is calculated by Db2 and does not count partitioned objects. Choose the value for DSMAx according to the impact that partitioning secondary indexes has on the number of data sets for those objects. Db2 defers closing and deallocating table spaces or indexes until the number of open data sets reaches the operating system limit or approaches the value that is specified by DSMAx.

Related concepts:
- How Db2 determines the initial value of DSMAx (Db2 Performance)

Related tasks:
- Managing the opening and closing of data sets (Db2 Performance)
- Evaluating the value of DSMAx (Db2 Performance)

Related information:
- 00C20113 (Db2 Codes)

**EDM STATEMENT CACHE field (EDMSTMTC subsystem parameter)**
The EDMSTMTC subsystem parameter determines the upper limit on the amount of EDM pool storage that is used for cached dynamic statements. The EDMSTMTC value is in kilobytes. This storage pool is located above the 2 GB bar.
Acceptable values: 5000 - 4194304
Default: 113386
Update: press ENTER on panel DSNTIPB
DSNZPxxx: DSN6SPRM EDMSTMTC

Notes:
1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTIPA1, based on values that you specify on panel DSNITPD.
   When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTIPA1, Db2 11 uses the Db2 11 field value. For more information, see Calculating storage requirements with the installation CLIST.

The Override column of panel DSNTIPC controls whether Db2 uses the calculated value:
   • To accept the calculated value, ensure that the Override column contains no value.
   • To override the calculated value, specify a value in the Override column.

You can use the SET SYSPARM command to change this value. If the amount of storage that is used exceeds the value of EDMSTMTC, Db2 reduces the amount of storage to the value of EDMSTMTC.

Storage for cached dynamic statements is allocated as it is needed. When the EDMSTMTC value is reached, and new statements are added to the dynamic statement cache, statements that are already in the cache and are not currently in use are removed, to keep the storage that is in use below the EDMSTMTC value.

Related tasks:
Calculating storage requirements with the installation CLIST
Calculating EDM pool sizes

Related reference:
[SET SYSPARM (Db2) (Db2 Commands)]

EDM DBD CACHE field (EDMDBDC subsystem parameter)
The EDMDBDC subsystem parameter determines the maximum amount of EDM storage, in kilobytes, that is to be used for DBDs. This storage pool is located above the 2 GB bar.

Acceptable values: 5000 - 4194304
Default: 23400
Update: press ENTER on panel DSNTIPB
DSNZPxxx: DSN6SPRM EDMDBDC

Notes:
1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTIPA1, based on values that you specify on panel DSNITPD.
   When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTIPA1, Db2 11 uses the Db2 11 field value. For more information, see Calculating storage requirements with the installation CLIST.
The *Override* column of panel DSNTIPC controls whether Db2 uses the calculated value:

- To accept the calculated value, ensure that the *Override* column contains no value.
- To override the calculated value, specify a value in the *Override* column.

You can use the SET SYSPARM command to change this value. If the amount of storage that is used exceeds the value of EDMDBDC, the Db2 database manager reduces the amount of storage to the value of EDMDBDC.

**Related tasks:**

- Calculating EDM pool sizes

**Related reference:**


**EDM SKELETON POOL SIZE field (EDM_SKELETON_POOL subsystem parameter)**

The EDM_SKELETON_POOL subsystem parameter determines the maximum size in kilobytes, that is to be used for the EDM skeleton pool. This storage is located above the 2 GB bar.

- Acceptable values: 5120 - 4194304
- Default: 51200
- Update: press ENTER on panel DSNTPB
- DSNZPxxx: DSN6SPRM EDM_SKELETON_POOL

**Notes:**

1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTIP1, based on values that you specify on panel DSNTPD.
   
   When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTIP1, Db2 uses the Db2 11 field value. For more information, see [Calculating storage requirements with the installation CLIST](https://www.ibm.com/support/knowledgecenter/en/SSD71H_11.1.0/com.ibm.db2.luw.itm.doc/doc/csy/mem_install.html)

The *Override* column of panel DSNTIPC controls whether Db2 uses the calculated value:

- To accept the calculated value, ensure that the *Override* column contains no value.
- To override the calculated value, specify a value in the *Override* column.

You can change this value by using the SET SYSPARM command. If the amount of storage that is used exceeds the value of EDM_SKELETON_POOL, Db2 reduces the amount of storage to the value of EDM_SKELETON_POOL.

**Related tasks:**

- Calculating EDM pool sizes

**Related reference:**


**BUFFER POOL SIZE field**

The value of the BUFFER POOL SIZE field is the buffer pool size that is calculated by the CLIST.
Notes:
1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTPA1, based on values that you specify on panel DSNITPD.

When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTPA1, Db2 12 uses the Db2 11 field value. For more information, see Calculating storage requirements with the installation CLIST.

The Override column of panel DSNTIPC controls whether Db2 uses the calculated value:

- To accept the calculated value, ensure that the Override column contains no value.
- To override the calculated value, specify a value in the Override column.

This field is protected and cannot be changed during update processing. If you want to change the size of a buffer pool, you must use the command ALTER BUFFERPOOL.

Related reference:
- ALTER BUFFERPOOL (Db2) (Db2 Commands)

SORT POOL SIZE field (SRTPPOOL subsystem parameter)
The SRTPPOOL subsystem parameter specifies the amount of storage in kilobytes that is needed for the sort pool.

Acceptable values: 240 - 128000
Default: 1000
Update: press Enter twice on panel DSNTIPB
DSNZPxxx: DSN6SPRM SRTPPOOL

Notes:
1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTPA1, based on values that you specify on panel DSNITPD.

When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTPA1, Db2 12 uses the Db2 11 field value. For more information, see Calculating storage requirements with the installation CLIST.

The Override column of panel DSNTIPC controls whether Db2 uses the calculated value:

- To accept the calculated value, ensure that the Override column contains no value.
- To override the calculated value, specify a value in the Override column.

If you decide to change this field, estimate the sort pool value by using the following formula:

\[ 32000 \times (16 + \text{sort key length} + \text{sort data length}) \]
Related tasks:

- Improving the performance of sort processing (Db2 Performance)
- Choosing buffer pool sizes (Db2 Performance)

MAX IN-MEMORY SORT SIZE field (MAXSORT_IN_MEMORY subsystem parameter)

The MAXSORT_IN_MEMORY subsystem parameter specifies the maximum allocation of storage in kilobytes for a query that contains an ORDER BY clause, a GROUP BY clause, or both. The storage is allocated only during the processing of the query. Increasing the value in this field can improve performance of such queries but might require large amounts of real storage when several such queries run simultaneously.

Acceptable values: 1000 to the value specified in SORT POOL SIZE, whichever is larger
Default: 1000
Update: press Enter twice on panel DSNTPB
DSNZPxxx: DSN6SPRM MAXSORT_IN_MEMORY

Notes:

1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTPA1, based on values that you specify on panel DSNITPD.

When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTPA1, Db2 12 uses the Db2 11 field value. For more information, see Calculating storage requirements with the installation CLIST

The Override column of panel DSNTPC controls whether Db2 uses the calculated value:

- To accept the calculated value, ensure that the Override column contains no value.
- To override the calculated value, specify a value in the Override column.

Related reference:

SORT POOL SIZE field (SRTPool subsystem parameter)

RID POOL SIZE field (MAXRBLK subsystem parameter)

The MAXRBLK subsystem parameter specifies the amount of storage in kilobytes that is needed for the RID pool.

Acceptable values: 0, 128 - 2000000
Default: 400000
Update: press Enter twice on panel DSNTPB
DSNZPxxx: DSN6SPRM MAXRBLK

Notes:

1. Db2 calculates the value when you run the installation CLIST with INSTALL for INSTALL TYPE on panel DSNTPA1, based on values that you specify on panel DSNITPD.

When you run the installation CLIST with MIGRATE for INSTALL TYPE on panel DSNTPA1, Db2 12 uses the Db2 11 field value. For more information, see Calculating storage requirements with the installation CLIST
The Override column of panel DSNTIPC controls whether Db2 uses the calculated value:

- To accept the calculated value, ensure that the Override column contains no value.
- To override the calculated value, specify a value in the Override column.

If you decide to change this field, estimate the storage that is required for the RID pool by using the following formula:

\[ \text{Number of concurrent RID processing activities} \times \text{average number of RIDs} \times 2 \times 8 \text{(bytes per RID)} \]

Choosing 0 disables the use of the RID pool. In this case, Db2 does not use access paths or join methods that depend on RID pool storage.

This storage is located above the 2 GB bar.

**Related tasks:**


**Related reference:**


**CLIST messages fields for panel DSNTIPC**

The CLIST messages fields display storage sizes that are calculated by the CLIST. These fields are protected and cannot be changed by the user.

Acceptable values: none
Default: none
Update: run CLIST again
DSNZPxxx: none

**Storage messages fields**

The storage messages fields indicate the results of the calculations described on panel DSNTIPC. These fields are protected and cannot be changed by a user.

Acceptable values: none
Default: none
Update: run CLIST again
DSNZPxxx: none

**DSNTIPC1: CLIST calculations panel 2**

The DSNTIPC1 panel displays CLIST messages that indicate a variety of calculations. This panel is displayed only when you run the CLIST in INSTALL mode.
CLIST messages fields for panel DSNTIPC1

The messages in these fields might be used depending on how many unique volume names were supplied on installation panel DSNTIP A2. These fields are protected and cannot be changed.

Acceptable values: none
Default: none
Update: run CLIST again
DSNZPxxx: none

CLIST panel session output

As output, the CLIST produces several data sets and a data set member.

Specifically, the CLIST panel session produces the following data sets and members:

prefix.SDSNSAMP
The existing data set contains a new member, if specified, that contains the resulting parameter values from the session.

prefix.NEW.SDSNSAMP
A new data set that contains the edited JCL with the values that you entered on the panels.

prefix.NEW.SDSNTEMP
A new data set that contains tailored CLISTs for input to job DSNTIJVC, which is run during installation or migration.

The following data sets are also created if you specify the use of z/OSMF for installation or migration. (If you run the CLIST to generate artifacts for provisioning Db2 subsystems, only these data sets are generated. For more information, see Creating artifacts for provisioning Db2 subsystems by running the installation CLIST.)
A new data set that contains the z/OSMF workflow definitions

A new data set contains the JCL templates for z/OSMF

You can use job DSNTIJVC to combine the CLISTs into a common data set.

Db2 performs validity checking of the values that you enter during the panel sessions. If you receive an ISPF error message, press the HELP key for additional information.

The following figure illustrates by examples how the CLIST works during installation.

The following figure illustrates how the CLIST works during migration.
Installation and migration jobs that the CLIST tailors

The installation CLIST uses the values that you specify on the installation panels to tailor and load the installation or migration jobs. Each job is composed of one or more JCL procedures or job steps.

The CLIST loads each job as a separate member of the newly created prefix.NEW.SDSNSAMP. Before you run any of these jobs, however, you might want to perform some editing that the CLIST does not do. If disk allocation is completely controlled by SMS for your installation, verify that the input to IDCAMS from the installation jobs does not conflict with the requirements for SMS.

The tailored jobs are described in the procedures for [Installing Db2] and [Migrating Db2 to Db2 12]. Recovery information is provided, along with a description of each job. Unless otherwise stated in the job description, a return code of 0 or 4 from any of the jobs indicates successful completion. Some of the jobs contain statements

---

Figure 76. Examples of input to and output from the installation CLIST during migration
that could fail without causing the job to fail. For instance, delete commands for
data sets, drop statements for SQL objects, and stop commands could fail when
you first run a job because the data sets or objects do not exist. Unless otherwise
stated, you can ignore these failures. The statements are needed to allow you to
rerun the job (if necessary) without performing the deletes, drops, and stops
manually; they are merely for cleanup or initialization processing. If a job fails,
follow the instructions that are provided in the recovery information for the job.

Which jobs you edit depends on the task that you are performing: installation,
migration, or update. In data sharing environments, you edit different jobs
depending on the data sharing function: group, member, or enable. The installation
CLIST tailors a different set of jobs for each task.

**Jobs for installing Db2 12**

If you are installing Db2 12, the CLIST tailors the following jobs:

- **DSNADMZW**
- **DSNTEJ0**
- **DSNTEJ1**
- **DSNTEJ1L**
- **DSNTEJ1P**
- **DSNTEJ1S**
- **DSNTEJIT**
- **DSNTEJ1U**
- **DSNTEJ2A**
- **DSNTEJ2C**
- **DSNTEJ2D**

If you specified a default WLM environment name in the WLM ENVIRONMENT
field of installation panel DSNTIPX, the CLIST edits DSNTEJ2U, DSNTEJ6O,
DSNTEJ6R, DSNTEJ6V, and DSNTEJ6W.

If you activated DDF and specified a value in the REMOTE LOCATION field of
panel DSNTIPY, the CLIST also tailors job DSNTEJ6.

If you activated DDF and specified a default WLM environment name in the WLM
ENVIRONMENT field of installation panel DSNTIPX, the CLIST edits DSNTEJ6Z,
DSNTEJ63, DSNTEJ64, DSNTEJ65, and DSNTEJ66.

If you activated DDF, specified a value in the REMOTE LOCATION field of panel
DSNTIPY, and specified a default WLM environment name in the WLM
ENVIRONMENT field of installation panel DSNTIPX, the CLIST edits DSNTEJ6D,
DSNTEJ6P, DSNTEJ65, DSNTEJ6T, DSNTEJ61, and DSNTEJ62.

If CICS is selected, the CLIST edits DSNTEJ5A, DSNTEJ5C, and DSNTEJ5P.

If you are using data sharing, the CLIST edits DSNTIJGF and DSNTIJFT.

The installation CLIST tailors the DSNHC, DSNH, DSNU, and DSNEMC01 CLISTs
for installation.
When SYSINSTL is specified on panel DSNTIPG, the installation CLIST performs the following additional changes:

- All BIND statements specify the package owner as entered in the INSTALL PKG OWNER field.
- All dynamic SQL streams begin with the SET CURRENT SQLID 'SYSINSTL' statement.
- All CREATE TABLE and CREATE INDEX statements are stripped of the creator name and preceded (as a group) by a SET CURRENT SCHEMA statement to indicate the creator.
- The following notice is added to the prolog of each affected job. Itemized statements are individually included or excluded, as applicable:

```c++
//* NOTICE: This job must be run by a user ID with DB2 Installation
//* System Operator authority because it has been customized
//* as follows for performing DB2 installation or migration
//* without DB2 System Administrator authority:
//* - Dynamic SQL processing is performed with the current
//* SQL ID set to SYSINSTL
//* - DSNTIAD runs with PLAN(DSNTIAIN)
//* - Creator IDs are removed from create statements for
//* tables and indexes
//* - SET CURRENT SCHEMA statements are added to provide
//* the necessary creator IDs for these objects
```

Additional tailoring is performed on job DSNTIJTM for SYSINSTL. For example, the job prepares and binds DSNTIAD, a program for processing executable dynamic SQL statements. It uses a version-specific naming convention of DSNTIAInn, where nn is 12, 11, or 10, to bind a package and plan for DSNTIAD. When configured for SYSINSTL, job DSNTIJTM binds an additional DSNTIAD package and plan, both named DSNTIAIN. No access is granted on the DSNTIAIN package or plan.

**Jobs for migrating to Db2 12**

If you are migrating to Db2 12, the CLIST tailors the following jobs:

- DSNADMZW
- DSNTEJxx
- DSNTEAxx
- DSNTECxx
- DSNTEExxx
- DSNTEJY
- DSNTEJZ
- DSNTIJCI
- DSNTIJCV
- DSNTIJEX
- DSNTIJFC
- DSNTIJG
- DSNTIJMV
- DSNTIJNC
- DSNTIJNH
- DSNTIJPD
- DSNTIJRC
- DSNTIJSD
- DSNTIJTC
- DSNTIJTM
- DSNTIJUZ
- DSNTIJVA
- DSNTIJUM
- DSNTIJUL
- DSNTIJVC

The installation CLIST tailors the DSNHC, DSNH, DSNU, and DSNEMC01 CLISTs for migration.

**Important:**

1. The DSNTEJxx jobs are for use only after new function is activated.
Job for activating Db2 12 function levels

If you are activating function level 500 or higher, the CLIST tailors the following jobs: DSNTIJEN, DSNTIJNF, DSNTIJES, DSNTIJCS, DSNTIEJxx, and DSNTIJNH.

Jobs for updating a Db2 12 installation

If you are updating, the CLIST tailors any of the following jobs based on the values that you update:

- DSNTIJUZ
- DSNTIJUA
- DSNTIJUM
- DSNTIJUL

Related information:

- DSNT489I (Db2 Messages)

Tailoring the installation jobs

You might want to tailor several items in the installation jobs, such as data set names or library names, to meet the requirements of your site.

About this task

The changes that are described below are general; that is, they apply to all the jobs processed by the CLIST.

Before you begin editing, you might want to print or back up the jobs. You can print the JCL for these jobs by using IEBPTPCH or any other print facility that is available at your site.

Procedure

To prepare the installation jobs for your site:

1. Tailor the jobs to suit the needs of your site. You should edit the jobs to conform to any unique requirements that you might have. Also, you might want to make any minor JCL changes for items that the ISPF panels did not handle.

2. If you specified volume serial numbers on installation panel DSNTIPA3, examine the volume serial numbers that are used in the various jobs. The volume serial number fields of installation panel DSNTIPA3 enable you to specify up to five volumes for the data sets that are defined during installation or migration. If you want to use more than five volumes, specify them before continuing with installation or migration tasks.

   If you specified volume serial numbers on installation panel DSNTIPA3, the DSNTINST CLIST spreads the data sets across these volumes. If you use explicit volume serials instead of SMS, adding more volumes to provide more separation of data sets can help improve system performance and recoverability. Many of the log data sets are large and easy to place on separate volumes. The CLIST produces a series of messages that estimate space for each type of data set that is created during the installation or migration process.

3. Edit the DSNH CLIST if needed. The DSNH CLIST allows you to precompile, compile, prelink-edit, link-edit, bind, and run an application by issuing a single
command. You might need to edit the DSNH CLIST to change values for some of the entries. Verify that DSNH keyword parameters for all Db2-supported compilers that your applications use.

a. Check the default data set names for the licensed programs that you have installed. These defaults are in the parameter definitions at the beginning of each program. If the names and prefixes are not correct for your site, change them.

b. Check default library names. If the names and prefixes are not correct for your site, change them. Ensure that the data sets exist and are cataloged for BLIB, CLIB, LLIB, and PLIB. When the DSNH CLIST runs, it creates DBRMLIB and LOAD data sets if they do not already exist. The DBRMLIB data set is created only if the DBRMLIB(DEFAULT) is set. The following are the default library names:

- BLIB(NONE)
- DBRMLIB(DEFAULT) - The DBRM library must be allocated exclusively when the precompiler writes to it.

**Recommendation**: Set up a temporary library, or one per user, rather than trying to share libraries.

- CLIB(NONE)
- LLIB(NONE)
- LOAD(RUNLIB.LOAD) - This library is allocated exclusively when it is being written.

**Recommendation**: Set up a temporary library, or one per user, rather than trying to share libraries.

- PLIB(NONE)

c. Check default processor options. If you prefer other default options, change them. The default processor options follow:

- CICSOPT(NONE)
- COPTION(NONE)  LOPTION(NONE)  PASS(DEFAULT)

**d.** Check print and work space defaults. If the default allocation sizes are not acceptable for your site, change them. The print and work space defaults follow:

- PSECSPAC(20)  WSECSPAC(20)
- PSPACE(20)  WSPACE(20)
- WORKUNIT(DEFAULT)

4. Examine the data set names for other products. Many data set names for other products appear in the jobs. These names are shown in the table on panel DSNTPU. Change them if they are different at your site.

**Related reference:**

- DSNTPA3: Data parameters panel
- DSNTPU: Host language data set names panel
- DSNH (TSO CLIST) (Db2 Commands)
Creating artifacts for provisioning Db2 subsystems by running the installation CLIST

You can run the installation CLIST to create artifacts for provisioning non-data-sharing Db2 subsystems.

Before you begin

Apply the PTF for APAR PH09857.

About this task

With IBM Cloud™ Provisioning and Management for z/OS, you can create services for provisioning non-data-sharing Db2 for z/OS subsystems. Certain artifacts are needed for creating the services, and one way to create the artifacts is by running the Db2 installation CLIST.

For an overview of process for creating services, see IBM Cloud Provisioning and Management for z/OS.

Tip: The CLIST approach supports more customization for the provisioned Db2 subsystem. However, you can also create artifacts for provisioning Db2 in a standard configuration by downloading and customizing the sample Db2 software services template. For more information, see Creating services for provisioning Db2 for z/OS subsystems (DBaaS).

Procedure

To create artifacts for services that provision Db2 subsystems by using running the installation CLIST, complete the following steps:

1. Run the installation CLIST and specify the values in the following steps. For general instructions for running the CLIST, see steps 1-6 in Tailoring Db2 12 installation and migration jobs with the CLIST.

2. In the CLIST panel session, specify on panel DSNTIPA1 that you are preparing artifacts for provisioning Db2 subsystems by specify the following options:

   - USE Z/OSMF WORKFLOW ==> YES
   - INSTALL TYPE ===> PROVISION
   - DATA SHARING ===> NO
   - Set the PREFIX and SUFFIX fields to the values that you used when you allocated the Db2 libraries by using job DSNALLOC.
   - INPUT MEMBER NAME ===> DSNTIDXC The default name is DSNTIDXC. You can also specify your own name. However, do not specify DSNTIDXA or DSNTIDXB.
   - OUTPUT MEMBER NAME ===> Specify a name other than DSNTIDXA, DSNTIDXB, or DSNTIDXC.

   When you specify these options, the PROVISIONING PREFERENCES AND DATA SET NAMES panel DSNTIPAP is shown next in the CLIST panel session.

3. On panel DSNTIPAP, specify the preferences for provisioning Db2 subsystems. For details, see DSNTIPAP: Provision DB2 - provisioning preferences and data set names.
4. On panel DSNTIPG, specify the user ID that executes the provisioning workflow in the PROVISION EXEC ID field. This value populates the AGEXECID input variable in DSNTIVIN.

5. Complete the rest of the CLIST panel session, as described in The Db2 installation CLIST panel session. The CLIST panel session is modified for the provisioning process. For example, certain z/OSMF workflow steps are force selected or deselected, some fields are preset with values generated in the provisioning process and protected, and all fields that are associated with z/OSMF variables are force selected.

Results

The CLIST generates the following output data sets. The generated z/OSMF artifacts are always written to the zFS path specified in the field PATH FOR Z/OSMF ARTIFACTS on panel DSNTIPM1. This field is mandatory for provisioning.

prefix.NEW.SDSNMFSA
A new data set that contains the z/OSMF variable input and workflow definitions, and the following new z/OSMF files that provide services for the provisioned Db2 subsystems.

- ACTIONS - Action definition file describing the actions that can be performed against the provisioned Db2 subsystem.
- DSNDDDF - Db2 Display DDF workflow
- DSNDEPRV - Deprovision workflow
- DSNOPTFT - Enable optional features workflow
- DSNSTART - Start Db2 workflow
- DSNSTOP - Stop Db2 workflow

prefix.NEW.SDSNSAM2
The JCL templates for z/OSMF.

The CLIST generates the following mandatory steps for provisioning stand-alone Db2 subsystems. These new steps are added at the beginning of the DSNTIWIN workflow, which is the workflow generated when installing a stand-alone Db2 subsystem.

1. s00INSSS — Instantiate and validate the Db2 subsystem identifier ssid. (DSNTRSSN)
2. s00SETVP — Set values for variables whose values are generated at provisioning time. (setVariable)
3. s00PORT — Acquire Db2 ports from the network resource pool. (REST API)
4. s00ALIAS — Define aliases for the instantiated Db2 system. (DSNTIJDA)
5. s00PPROC — Allocate private procedure library for the instantiated Db2 system. (DSNTIJPP)
6. s00SETVL — Assign values to BSDS, active and archive logs variables. (setVariable)
7. s00IRLM — Assign IRLM SSID based on the instantiated Db2 system SSID. (Inline shell-JCL)
8. s00DEFSS — Define the instantiated Db2 and IRLM systems to z/OS. (DSNTIJMD)
9. stepJTU — Create and grant usage of Db2 storage group for user data (DSNTIJTU)
The CLIST also generates the following optional steps, which are appended at the end of the DSNTIWIN workflow. These steps are not displayed on any installation panel for a user to select for execution. Instead, they can be selected for execution at provisioning time.

1. stepJRP — Enable Db2 REST services on the provisioned system. (DSNTIJRP)
2. stepODBC — Install and verify the ODBC for the Db2 subsystem. (DSNTIJCL, DSNTEJ8)
3. stepJDBC — Install and verify JDBC for the Db2 subsystem. (multiple sub steps)

**Related tasks:**

Creating services for provisioning Db2 for z/OS subsystems (DBaaS)

## Updating subsystem parameter and application default values

You can dynamically update the value of some subsystem parameters while Db2 is running. For other subsystem parameters, you must stop and restart Db2 to change the values.

### Before you begin

**Important:** Changes to subsystem parameters affect the entire subsystem. Carefully evaluate and plan for the impact of any changes to these values. Also, follow any advice that is included in the documentation for the parameter.

Determine whether you will update the subsystem parameters values online or offline. Not all parameters can be updated online. To check whether a particular subsystem parameter can be updated online, see Directory of subsystem parameters, panel fields, and application default values.

### About this task

**Restriction:** The following procedure cannot be used for some subsystem parameters that define characteristics of the Db2 catalog, directory, BSDS, and logs or for the DDF subsystem parameter. For these parameters, follow the instructions in Updating subsystem parameters that cannot be updated by using DSNTIPB.

### Procedure

To update subsystem parameter values, complete the following steps:

1. Invoke the CLIST:
   a. Check that your TSO logon region size is sufficient. Usually 2 MB is enough.
   b. Invoke ISPF.
   c. Select option 6 on the main ISPF panel.
   d. Invoke the installation CLIST by entering one of the following specifications:
      - EXEC 'prefix.SDSNCLST(DSNTINST)'
      - If you want to receive the messages that trace the progress of the CLIST, use the following specification:
        EXEC 'prefix.SDSNCLST(DSNTINST) 'CONTROL(LIST)'
      - If you want to use the CLIST but skip the verification of output data sets, use the following specification:
By default, the CLIST verifies the following information when installation panels DSNTIPT and DSNTIPA1 are displayed:
- The CLIST verifies that the TEMP CLIST LIBRARY and SAMPLE LIBRARY that you specify on DSNTIPT can be allocated and opened for output.
- If you specify a value for OUTPUT MEMBER NAME on DSNTIPA1, the CLIST verifies that the data set prefix.SDSNSAMP can be allocated and opened for output.

The CLIST reads a set of default values and displays them on the panels. The values can be the original default values that are supplied by IBM, values that you specified in a previous run of the CLIST, or values that were generated when you ran the DSNTXAZP tool.

2. On installation panel DSNTIPA1, in field 1 INSTALL TYPE, specify UPDATE and press Enter.

3. On installation panel DSNTIPT, in field 2 SAMPLE LIBRARY, specify the output SDSNSAMP data set and press Enter. Panel DSNTIPB is displayed.

4. On panel DSNTIPB, specify the installation panel that you want to update and press Enter.

5. Change the values on the panel as needed, and press Enter to return to panel DSNTIPB.

6. Optional: Repeat steps 4 - 5 until you specify all of the updates that you want.

7. On panel DSNTIPB, press ENTER again to complete the update process. Depending on the values that you change, this process produces any of the following jobs with the new values: DSNTIJUZ, DSNTIJUA, and DSNITJUL.

The default SQL processing level (SQLLEVEL) for the precompiler/coprocessor can be changed anytime, such as after activating a higher Db2 function level. For more information, see [PRECOMPILES SQL LEVEL field](SQLLEVEL.DECP value). For updates, the Db2 installation CLIST generates job DSNTIJUL only for changes to any of the following parameters:
- DB2 LOCATION NAME field
- DB2 VTAM LUNAME field
- DB2 GENERIC LUNAME field
- DB2 VTAM PASSWORD field
- DB2 TCP/IP IPNAME field
- TCP/IP PORT field
- SECURE PORT field
- RESYNC PORT field

This process produces a new DSNTIJUZ job with the new values; it also saves these values in the file that is specified in the OUTPUT MEMBER NAME field on panel DSNTIPA1. For more information, see OUTPUT MEMBER NAME field.
8. Run job DSNTIJUZ. Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This action ensures that members DSNFYSYP, DSNJARVP, DSNJLOGP, DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```plaintext
SET BOUNDARY (GLOBAL).
LIST DODEF ('SMPTLIB').
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions.

For more information, see Job DSNTIJUZ: define the Db2 data-only subsystem parameter module.

9. If you modified application programming default values, run the generated DSNTIJUA job. Job DSNTIJUA defines the Db2 data-only application defaults module.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUA to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUA so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZQ. This action ensures that member DSNARIB is linked with the he application defaults module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```plaintext
SET BOUNDARY (GLOBAL).
LIST DODEF ('SMPTLIB').
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

For more information, see Job DSNTIJUA: define data-only application defaults module.

10. If you updated the following DDF related values, run the generated DSNTIJUL job.

- **DB2 LOCATION NAME field**
- **DB2 VTAM LUNAME field**
Job DSNTIJUL runs the Db2 change log inventory (DSNJU003) to update DDF-related information in the BSDS. For more information, see Job DSNTIJUL: update DDF related BSDS information.

11. Use one of the following methods to tell the subsystem to begin using the new load module:
   - For online updates, issue the SET SYSPARM command:
     ```
     SET SYSPARM LOAD(load-module-name)
     ```
     where load-module-name is the same as the OUTPUT MEMBER NAME field value that was specified on the DSNTIPA1 panel.
     If you want to specify the load module name that is used during Db2 startup, you can issue the following command:
     ```
     SET SYSPARM RELOAD
     ```
   - For offline updates, stop and restart Db2 with the new subsystem parameter load module.

Related tasks:
- Optimizing subsystem parameters (Db2 Performance)
- Optimizing subsystem parameters for SQL statements by using profiles (Db2 Performance)

Related reference:
- DSNTIPA1 or DSNTILA1: Main panel
- DSNTIPT: Db2 data set names panel
- DSNTIPB: Update selection menu panel
- SET SYSPARM (Db2) (Db2 Commands)

Related information:
- Subsystem parameters (Introduction to Db2 for z/OS)
- Application programming default values (Introduction to Db2 for z/OS)
- Directory of subsystem parameters, panel fields, and application default values
- Bootstrap data set (Introduction to Db2 for z/OS)

**DSNTIPB: Update selection menu panel**

During the update process, you can use the DSNTIPB panel to access all the installation panels and view the values that you specified during installation or migration.

Parameters that you can update are highlighted. Panels containing fields that cannot be updated are marked with an asterisk.
**Panels to update field**

From the DSNTIPB panel, you can select a family of parameters to display and update. Some parameters cannot be updated.

On the command line, enter a number to select the family of parameters that you want to update. These numbers correspond to the installation panels in the following table.

**Table 41. Panel names and identifiers, by DSNTIPB panel to update field option values**

<table>
<thead>
<tr>
<th>Option Value</th>
<th>Panel Name and Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DSNTIPA2: Db2 catalog and directory panel</td>
</tr>
<tr>
<td>2</td>
<td>DSNTIPA3: Data parameters panel</td>
</tr>
<tr>
<td>3</td>
<td>DSNTIPA4: BSDS and active log panel</td>
</tr>
<tr>
<td>4</td>
<td>DSNTIPK: Define group or member panel</td>
</tr>
<tr>
<td>5</td>
<td>DSNTIP00: Activate a function level panel</td>
</tr>
<tr>
<td>6</td>
<td>DSNTIPC: Installation preferences panel</td>
</tr>
<tr>
<td>7</td>
<td>DSNTIPG1: Installation preferences panel 2</td>
</tr>
<tr>
<td>8</td>
<td>DSNTIPH: System resource data set names panel</td>
</tr>
<tr>
<td>9</td>
<td>DSNTIPT: Db2 data set names panel</td>
</tr>
<tr>
<td>10</td>
<td>DSNTIPU: Host language data set names panel</td>
</tr>
</tbody>
</table>
Table 41. Panel names and identifiers, by DSNTIPB panel to update field option values (continued)

<table>
<thead>
<tr>
<th>Option Value</th>
<th>Panel Name and Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>DSNTIPU1: Host language data set names panel 2</td>
</tr>
<tr>
<td>12</td>
<td>DSNTI PW: Other data set names panel</td>
</tr>
<tr>
<td>13</td>
<td>DSNTIPD: Sizes panel 1</td>
</tr>
<tr>
<td>14</td>
<td>DSNTIP7: SQL OBJECT DEFAULTS PANEL 1</td>
</tr>
<tr>
<td>15</td>
<td>DSNTIP71: SQL OBJECT DEFAULTS PANEL 2</td>
</tr>
<tr>
<td>16</td>
<td>DSNTIP72: SQL OBJECT DEFAULTS PANEL 3</td>
</tr>
<tr>
<td>17</td>
<td>DSNTIP9: Work file database panel</td>
</tr>
<tr>
<td>18</td>
<td>DSNTIP91: Work file database panel 2</td>
</tr>
<tr>
<td>19</td>
<td>DSNTIPE: Thread management panel 1</td>
</tr>
<tr>
<td>20</td>
<td>DSNTIE1: Thread management panel</td>
</tr>
<tr>
<td>21</td>
<td>DSNTIP1: Buffer pool sizes panel 1</td>
</tr>
<tr>
<td>22</td>
<td>DSNTIP2: Buffer pool sizes panel 2</td>
</tr>
<tr>
<td>23</td>
<td>DSNTIPN: Tracing parameters panel</td>
</tr>
<tr>
<td>24</td>
<td>DSNTIPO: Operator functions panel</td>
</tr>
<tr>
<td>25</td>
<td>DSNTIPO3: Default startup modules panel</td>
</tr>
<tr>
<td>26</td>
<td>DSNTIPO4: Resource Limit Facility panel</td>
</tr>
<tr>
<td>27</td>
<td>DSNTIPF: Application programming defaults panel 1</td>
</tr>
<tr>
<td>28</td>
<td>DSNTIP4: Application programming defaults panel 2</td>
</tr>
<tr>
<td>29</td>
<td>DSNTIP8: Performance and optimization panel</td>
</tr>
<tr>
<td>30</td>
<td>DSNTIP81: Performance and optimization panel 2</td>
</tr>
<tr>
<td>31</td>
<td>DSNTIP82: Query accelerator preferences panel</td>
</tr>
<tr>
<td>32</td>
<td>DSNTIPI: IRLM panel 1</td>
</tr>
<tr>
<td>33</td>
<td>DSNTIPJ: IRLM panel 2</td>
</tr>
<tr>
<td>34</td>
<td>DSNTIP6: Db2 utilities parameters panel 1</td>
</tr>
<tr>
<td>35</td>
<td>DSNTIP61: Db2 utilities parameters panel 2</td>
</tr>
<tr>
<td>36</td>
<td>DSNTIP62: Db2 utilities parameters panel 3</td>
</tr>
<tr>
<td>37</td>
<td>DSNTIP63: Db2 utilities parameters panel 4</td>
</tr>
<tr>
<td>38</td>
<td>DSNTIPP: Protection panel</td>
</tr>
<tr>
<td>39</td>
<td>DSNTIPP1: Protection panel 2</td>
</tr>
<tr>
<td>40</td>
<td>DSNTIPM: MVS PARMLIB updates panel</td>
</tr>
<tr>
<td>41</td>
<td>DSNTIIPL: Active log data set parameters</td>
</tr>
<tr>
<td>42</td>
<td>DSNTIPL1: Checkpoint parameters</td>
</tr>
<tr>
<td>43</td>
<td>DSNTIPA: Archive log data set parameters panel</td>
</tr>
<tr>
<td>44</td>
<td>DSNTIPS: Databases and spaces to start automatically panel</td>
</tr>
<tr>
<td>45</td>
<td>DSNTIPR: Distributed data facility panel 1</td>
</tr>
<tr>
<td>46</td>
<td>DSNTIP5: Distributed data facility panel 2</td>
</tr>
<tr>
<td>47</td>
<td>DSNTIPX: Routine parameters panel</td>
</tr>
<tr>
<td>48</td>
<td>DSNTIPX1: Db2 Java properties panel</td>
</tr>
</tbody>
</table>
Table 41. Panel names and identifiers, by DSNTIPB panel to update field option values (continued)

<table>
<thead>
<tr>
<th>Option Value</th>
<th>Panel Name and Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>DSNTIPR1: Parameters for Db2-supplied routines panel</td>
</tr>
<tr>
<td>50</td>
<td>DSNTIPZ: Data definition control support panel</td>
</tr>
<tr>
<td>51</td>
<td>DSNTIPY: Job editing panel</td>
</tr>
</tbody>
</table>

When the panel that you select is displayed, enter the new parameters. Press the Enter key to return to the update selection menu panel. Make another panel selection, or press Enter again to process. Press End to leave the update selection menu panel and return to the main panel (DNSTIPA1).

Related reference:
DSNTIPA1 or DSNTILA1: Main panel

Updating subsystem parameters that cannot be updated by using DSNTIPB

Some subsystem parameters cannot be updated by using panel DSNTIPB.

Procedure

To update subsystem parameters that you cannot update by using panel DSNTIPB:

- To update the CATALOG ALIAS and DEFINE CATALOG fields on DSNTIPA2, see [Changing the high-level qualifier for Db2 data sets (Db2 Administration Guide)](#). The CATALOG ALIAS parameter establishes an alias name for your ICF catalog. This name is also used as the high-level qualifier name for Db2 VSAM data sets. The DEFINE CATALOG parameter controls the creation of the ICF catalog.

- To update Db2 to use the distributed data facility (DDF), follow these steps:
  1. Go through the normal update process of running the CLIST to add DDF information to installation panel DSNTIPR.
  2. Run job DSNTIJUZ.
  3. Populate the CDB.
  4. Stop and start Db2.
  5. Bind or rebind these plans:

```sql
BIND PLAN(DSNESPCS) PKLIST(*.DSNESPCS.DSNESM68) ISOLATION(CS) ACTION(REPLACE)
BIND PLAN(DSNESPRR) PKLIST(*.DSNESPRR.DSNESM68) ISOLATION(UR) ACTION(REPLACE)
```

6. Start DDF if you specified COMMAND instead of AUTO as the DDF STARTUP OPTION on installation panel DSNTIPR.

- To change the data set sizes for the Db2 catalog and directory:
  1. Copy the catalog and directory table spaces.
2. Stop the table spaces or their databases.
3. Delete the data sets and redefine them, using VSAM commands.
4. Use the RECOVER utility to recover the catalog and directory to the new data sets.
5. Start the table spaces or databases again.

• To change from single to dual logging for the active log:
  1. Define the second copy of the log with a VSAM IDCAMS DEFINE statement. Refer to job DSNTIJIN, which contains the DEFINE statement for the first copy of the log.
  2. Run the DSNJU003 (change log inventory) utility. This adds the second copy of the log to the BSDS.
  3. Update the NUMBER OF COPIES field on installation panel DSNTIPFH from 1 to 2.
  4. Run job DSNTIJUZ to make the change effective.

• To move or expand the bootstrap data sets, use the IMPORT and EXPORT commands of access method services. The bootstrap data sets are accessed using JCL when Db2 starts.
• To access the log data sets, you can use stand-alone access macros or the IMPORT and EXPORT commands of access method service.
• To change the number of data sets for active logs, you can use the DSNJU003 utility.

Related tasks:

Populating the communications database for use with TCP/IP

Defining a new integrated catalog alias (Db2 Administration Guide)
YES

Enables modeling of query acceleration. The YES value applies only when the following requirements are also met:

- The value of the QUERY ACCELERATION subsystem parameter value is NONE, ENABLE, ENABLE WITH FAILBACK or ELIGIBLE.
- The CURRENT GET_ACCEL_ARCHIVE special register value is NO.
- The GET_ACCEL_ARCHIVE subsystem parameter value is NO

If other values are specified, Db2 tries to accelerate queries instead of doing accelerator modeling.

Only queries that Db2 deems eligible to run on an accelerator server are included in the accelerator-related fields of accounting trace IFCID 3.

Related tasks:

- Determining whether queries can benefit from acceleration (Db2 Performance)

Related reference:

- CURRENT QUERY ACCEL field (QUERY_ACCELERATION subsystem parameter)

- CURRENT GET_ACCEL_ARCHIVE (Db2 SQL)

- GET ACCEL_ARCHIVE field (GET_ACCEL_ARCHIVE subsystem parameter)

**AUTH_COMPATIBILITY in macro DSN6SPRM**

The AUTH_COMPATIBILITY parameter specifies whether current-release of previous-release authorization rules are used for specific Db2 operations.

Db2 12 introduces this subsystem parameter. It takes effect only when the activated function level is function level 500 or higher.

Acceptable values: blank, SELECT_FOR_UNLOAD

Default: blank

DSNZPxxx: DSN6SPRM AUTH_COMPATIBILITY

Security parameter: Yes

**Null (blank)**

Indicates that AUTH_COMPATIBILITY has no effect and that there is no authorization check override.

**SELECT_FOR_UNLOAD**

- When this value is specified, the UNLOAD utility checks if the user has the SELECT privilege on the target table.
- When this value is not specified, the UNLOAD utility checks if the user has the UNLOAD privilege on the target table.

Related concepts:

- UNLOAD privilege for the UNLOAD utility (Db2 for z/OS What’s New?)

Related reference:

- UNLOAD (Db2 Utilities)
- Explicit table and view privileges (Managing Security)
CDDS_MODE in macro DSN6LOGP

The CDDS_MODE subsystem parameter specifies how a member of a Db2 data sharing group participates in a GDPS® Continuous Availability with zero data loss (GDPS Continuous Availability with zero data loss) environment.

Acceptable values: NONE, SOURCE, PROXY
Default: NONE
Data sharing scope: A setting of SOURCE or PROXY is valid only in a data sharing environment. All members of the data sharing group must have the same setting.
Online changeable: No
DSNZPxxx: DSN6LOGP.CDDS_MODE

NONE
The Db2 subsystem or data sharing group does not participate in a GDPS Continuous Availability with zero data loss environment. The subsystem or data sharing group member supports only replication of its own log. The data set that the CDDS_PREFIX parameter specifies can be defined locally but is not used.

SOURCE
The data sharing group that contains the member that is being installed is the source group in an implementation of the GDPS Continuous Availability with zero data loss solution. The source data sharing group supports replication of the logs and log data sets of its members by another data sharing group, which is the proxy group in the implementation of the GDPS Continuous Availability with zero data loss solution. The proxy group performs replication of tables in the source data sharing group that are enabled for changed data capture.

The source group must contain a key-sequenced data set with the name that is specified by the CDDS_PREFIX subsystem parameter. That data set is the compression dictionary data set (CDDS). When IFI READS calls are made to the proxy group to retrieve compressed log records, Db2 uses the CDDS to decompress the log records.

The location of the source group must be defined in the communications database of the proxy group.

PROXY
The data sharing group that contains the member that is being installed supports only replication of log data sets for the source data sharing group in an implementation of the GDPS Continuous Availability with zero data loss solution.

The location of the proxy group must be defined in the communications database of the source group.

Related tasks:
- Populating the communications database for use with TCP/IP
- Upgrading a Db2 11 GDPS Continuous Availability with zero data loss environment to a Db2 12 environment (Db2 Administration Guide)

Related reference:
- CDDS_PREFIX in macro DSN6LOGP

CDDS_PREFIX in macro DSN6LOGP

The CDDS_PREFIX subsystem parameter specifies the prefix for the name of the data set that contains dictionaries that Db2 uses to decompress compressed log
records when a proxy data sharing group in an implementation of the GDPS Continuous Availability with zero data loss (GDPS Continuous Availability with zero data loss) solution captures the log records on behalf of a source data sharing group. The data set is known as the compression dictionary data set (CDDS).

Acceptable values: 1 - 39 bytes that form a valid z/OS data set name prefix
Default: DSNCAT
Data sharing scope: All members of the data sharing group must have the same setting.
Online changeable: No
DSNZPxxx: DSN6LOGPCDDS_PREFIX

The data set name is formed by appending the string ‘.CDDS’ to the value of CDDS_PREFIX.

This subsystem parameter is used only if the Db2 data sharing group that contains a member that is being installed is part of an implementation of the GDPS Continuous Availability with zero data loss solution.

If CDDS_MODE is set to SOURCE or PROXY, the CDDS must be defined at the source data sharing group and mirrored through peer-to-peer remote copy (PPRC) at the proxy data sharing group.

Related tasks:
- Modifying Db2 for the GDPS Continuous Availability with zero data loss solution (Db2 Administration Guide)
- Upgrading a Db2 11 GDPS Continuous Availability with zero data loss environment to a Db2 12 environment (Db2 Administration Guide)

Related reference:
- CDDS_MODE in macro DSN6LOGP

COMCRIT in macro DSN6SPRM
Sets the Common Criteria environment, which requires that tables that you create (other than created global temporary tables, declared global temporary tables, and auxiliary tables) must have multilevel security.

Acceptable values: NO, YES
Default: NO
DSNZPxxx: DSN6SPRM COMCRIT
Security parameter: Yes

NO  The behavior of Db2 is unchanged.
YES

All tables that you create (other than created global temporary tables, declared global temporary tables, and auxiliary tables) must have multilevel security. If the AS SECURITY LABEL clause is missing from a table, an error occurs and the table is not created. Setting COMCRIT to YES causes some of the current installation and migration processes to fail. A value of YES also limits the number of concurrent sessions that are allowed for a primary authorization ID if a RACF custom field is defined for that ID.

You can change the value of COMCRIT online by using SET SYSPARM and you can audit COMCRIT with IFCID 0106.

Related concepts:
CHECK_FASTREPLICATION in macro DSN6SPRM

The CHECK_FASTREPLICATION parameter specifies the type of replication that DSS COPY uses to copy objects to shadow data sets when it is invoked by the Db2 CHECK utilities. The CHECK utilities can stipulate fast replication as NONE, PREFERRED, or REQUIRED.

Acceptable values: PREFERRED, REQUIRED, NONE
Default: REQUIRED
DSNZPxxx: DSN6SPRM.CHECK_FASTREPLICATION

PREFERRED
The CHECK utilities direct DSS COPY to use fast replication only if FlashCopy support is available.

REQUIRED
The CHECK utilities force DSS COPY to use fast replication, ensuring that object copies occur as quickly as possible. This reduces the opportunity for resource contention and unavailability. This option causes CHECK utilities to fail if FlashCopy cannot be used.

NONE
The CHECK utilities direct DSS COPY not to use fast replication.

Related concepts:

Subsystem parameters for refining DFSMSdss COPY operation with utilities (Db2 Utilities)

Related tasks:

Checking data consistency with the CHECK utilities (Managing Security)

DDF_COMPATIBILITY in macro DSN6FAC

Controls certain characteristics of a connection between a client application and a Db2 for z/OS data server.

Values can be specified in any combination and in any order. If more than one value is specified, the values must be separated by commas, and the entire set of values must be enclosed in parentheses. For example, any of the following specifications are valid:

DDF_COMPATIBILITY=DISABLE_IMPCAST_NJV

DDF_COMPATIBILITY=(SP_PARMS_NJV,SP_PARMS_JV)

DDF_COMPATIBILITY=(DISABLE_IMPCAST_NJV,DISABLE_IMPCAST_JV)

DDF_COMPATIBILITY=(DISABLE_IMPCAST_NJV,SP_PARMS_NJV,DISABLE_IMPCAST_JV)

You can specify the following values for DDF_COMPATIBILITY:

null
The DDF_COMPATIBILITY parameter has no effect and Db2 12 new function behavior is used for all applications.

DISABLE_IMPCAST_JV
Specifies that the Db2 for z/OS server disables implicit casting of input host
variables from numeric data types to string data types, or from string data
types to numeric data types, when the application is a Java client application
that uses the IBM Data Server Driver for JDBC and SQLJ.

If DISABLE_IMPCAST_JV is not specified, Db2 enables implicit casting of
input host variables from numeric data types to string data types, or from
string data types to numeric data types.

DISABLE_IMPCAST_JV is not needed and has no effect if the client JDBC
driver is at version 3.63 or 4.13, or later.

IDNTFY_V12_PRIOR_VER
Specifies that a Db2 12 server at any function level identifies itself to all remote
clients as a Db2 11 server in new-function mode.

Specify IDNTFY_V12_PRIOR_VER as a temporary solution if an error occurs
for a connection from a remote client or driver only because the server
identifies itself to the client or driver as a Db2 12 server.

IGNORE_TZ
Specifies that the time zone information is ignored during boundary error
checking for TIMESTAMP WITH TIME ZONE input variables when the target
data type is TIMESTAMP and the client driver is IBM Data Server Driver for
JDBC and SQLJ.

IGNORE_TZ is deprecated. In Db2 11, IGNORE_TZ is effective only if
application compatibility is set to V10R1.

IGNORE_TZ is not needed and has no effect if the client JDBC driver is at
version 3.63 or 4.13, or later.

SP_PARMS_NJV
Specifies that when a non-Java client application calls a Db2 for z/OS stored
procedure, Db2 returns output argument values with data types that match the
data types of the corresponding CALL statement arguments.

Exception: Db2 does not return output argument values with data types that
match the data types of the corresponding CALL statement arguments if any
of the following conditions is true:
• The non-Java client is Version 10.1 or later.
• The stored procedure uses a parameter data type that was introduced in
  DB2 10 (XML, TIMESTAMP WITH TIMEZONE, or TIMESTAMP with
  precision greater than 6).
• Db2 for z/OS is at function level V12R1M501 or later.

If either condition is true, Db2 returns output parameter values with data types that
match the data types of the parameters in the stored procedure definition.

If SP_PARMS_NJV is not specified, Db2 returns output parameter values with
data types that match the data types of the parameters in the stored procedure
definition.

SP_PARMS_NJV is deprecated. Db2 ignores this value if application
compatibility is set to V11R1 or later.

DISALLOW_SEL_INT
Specifies whether to disallow a UNION or UNION ALL as the outermost
from-clause of a SELECT INTO statement, which is invalid SQL syntax. Such
statements were inadvertently allowed before Db2 12. However, in Db2 12 they are
disallowed by default.
Acceptable values: YES, NO
Default: YES
Online changeable: YES
DSNZPxxx: DSN6SPRM DISALLOW_SEL_INT INTO UNION

YES
SELECT INTO statements with UNION or UNION ALL as the outermost from-clause are not supported. This setting enforces valid SQL syntax.

NO
SELECT INTO statements with UNION or UNION ALL as the outermost from-clause are accepted. This setting tolerates invalid SQL syntax.

Important: In Db2 12, the default value of DISALLOW_SEL_INT INTO UNION changes from NO to YES.

DISALLOW_SEL_INT INTO UNION is also deprecated. Although it remains supported, support is likely to be removed eventually. Identify any packages that use UNION or UNION ALL in the from-clause of a SELECT INTO statement and correct them as necessary.

You can identify affected packages while DISALLOW_SEL_INT INTO UNION is set to NO by binding suspected packages into a dummy collection ID with EXPLAIN(ONLY) and monitoring IFCID 0376 records. Use the following procedure:

1. Issue the following SQL statement to generate a list of BIND commands.

   ```sql
   SELECT 'BIND PACKAGE(DUMMYCOL) COPY(' ||
   COLLID || '. ' || NAME || ') ' ||
   CASE WHEN(VERSION<>'' ) THEN 'COPYVER(' || VERSION || ')' ' ELSE '' END ||
   'EXPLAIN(ONLY)' ||
   FROM SYSIBM.SYSPACKSTMT
   WHERE STATEMENT LIKE '%SELECT%INTO%UNION%';
   ```

   The statement generates output similar to the following result:

   ```sql
   BIND PACKAGE(DUMMYCOL) COPY(DSN_DEFAULT_COLLID_PLAY01.PLAY01) EXPLAIN(ONLY)
   ```

2. Copy the SELECT statement in a BIND job. If the result is longer than 72 characters, formatting is required.

3. Start and collect the IFCID 0376 trace.

4. Run the bind job that you created.

5. Stop the IFCID 0376 trace and analyze the output.

Related concepts:

- Application and SQL release incompatibilities (Db2 for z/OS What's New?)

**ENCRYPTION_KEYLABEL field in macro DSN6SPRM**

The ENCRYPTION_KEYLABEL subsystem parameter specifies the name of an ICSF key label to provide to DFSMS when using z/OS DFSMS data set encryption support to define encrypted data sets for the Db2 catalog, directories, and archive logs.

The setting is used to define encrypted data sets for the Db2 catalog, directory objects, user-defined indexes on the catalog, and archive logs only when the Db2 function level is V12R1M502 or higher.
The default setting of blank means Db2 does not provide the key label during allocation of the data sets associated with the system objects. If non-blank, ENCRYPTION_KEYLABEL must be set to a string from 1 - 64 bytes that identifies a protected data key in the ICSF key repository.

A change to this parameter does not take effect until you use the -SET SYSPARM command to bring it online. Even if you start or restart Db2 after changing the value, the change is not honored until you issue the -SET SYSPARM command. In Db2 data sharing, all members need to use the same setting. When changing the setting of ENCRYPTION_KEYLABEL for a data sharing group, make the change on all members before running the -SET SYSPARM command on any member.

**EN_PJSJ in macro DSN6SPRM**
Disables or enables dynamic index ANDing, which is also called pair-wise join processing, when star-join processing is enabled on Db2.

Acceptable values: OFF, ON
Default: OFF
DSNZPxxx: DSN6SPRM EN_PJSJ
Security parameter: No

**OFF**
Dynamic index ANDing is disabled.

**ON**
Dynamic index ANDing is enabled.

When dynamic index ANDing is enabled, it can improve the execution performance of qualified queries in the following cases:

- The fact table has indexes that provide matching for join predicates with some dimension tables having local filtering. The indexes on the fact table can be single-key indexes (but not limited).
- No multi-column index on the fact table is suitable for the existing star join method.

Setting EN_PJSJ to ON might provide more flexibility with index design on star schema fact tables when multi-column index design makes effective exploitation the star join access method difficult.

**Related concepts:**

[Pair-wise join access (JOIN_TYPE='P') (Db2 Performance)]
[Star join access (JOIN_TYPE='S') (Db2 Performance)]

**HONOR_KEEPDICTIONARY in macro DSN6SPRM**
Specifies whether Db2 honors the LOAD and REORG parameter KEEPDICTIONARY when tables are converted between basic row format and reordered row format.

Acceptable values: NO, YES
Default: NO
DSNZPxxx: DSN6SPRM HONOR_KEEPDICTIONARY
Security parameter: No

**NO**
Db2 ignores the LOAD and REORG parameter KEEPDICTIONARY when tables are converted between basic row format and reordered row format.

**YES**
Db2 honors the LOAD and REORG parameter KEEPDICTIONARY.
**IMPDSSIZE in macro DSN6SYSP**

Specifies the default maximum data set size (DSSIZE) in gigabytes, for implicitly created table spaces for base tables.

- **Acceptable values:** 1, 2, 4, 8, 16, 32, 64, 128, 256
- **Default:** 64 for SAP, otherwise 4
- **DSNZPxxx:** DSN6SYSP IMPDSSIZE
- **Security parameter:** No

1, 2, 4, 8, 16, 32, 64, 128, 256

Maximum data set size (DSSIZE) in gigabytes that Db2 uses for creating implicit base table spaces.

If you specify an SAP-specific input member (for example, DSNTIDXB) during installation or migration, the DSNTIJUZ job that is generated by the Db2 CLIST specifies IMPDSSIZE=64.

**Related concepts:**

- [Implicitly defined table spaces (Db2 Administration Guide)](#)

**IMPTKMOD in macro DSN6SYSP**

Specifies the default TRACKMOD option for CREATE TABLESPACE statements and implicitly created tables spaces, which specifies Db2 tracks changed pages of the base table space in the space map pages.

- **Acceptable values:** YES, NO
- **Default:** YES
- **DSNZPxxx:** DSN6SYSP IMPTKMOD
- **Security parameter:** No

YES

Db2 tracks changed pages of the base table space in the space map pages. This tracking is intended to improve the performance of incremental image copies. YES is the default value.

NO

Db2 does not track changed pages in the space map pages. Db2 uses the LRSN value in each page to determine whether a page has been changed.

If you specify an SAP-specific input member (for example, DSNTIDXB) during installation or migration, the DSNTIJUZ job that is generated by the Db2 CLIST specifies IMPTKMOD=NO.

The setting is not used for LOB or XML table spaces.

**Related concepts:**

- [Control of tracking updates to reduce coupling facility overhead (Db2 Data Sharing Planning and Administration)](#)

**Related reference:**

- [CREATE TABLESPACE (Db2 SQL)](#)

**INLISTP in macro DSN6SPRM**

Specifies the maximum number of elements in an IN-list for certain IN predicate optimizations to occur.

- **Acceptable values:** 1 - 5000, 0
- **Default:** 50
DSNZPxxx: DSN6SPRM INLISTP
Security parameter: No

1 - 5000

Db2 optimizes for an IN predicate with the specified number of values, with the following optimizations:

- Pushes the IN predicate down from the parent query block into the materialized table expression.
- Moves a correlated IN predicate in a subquery that is generated by transitive closure up to the parent query block.

0  Disables optimization for in predicates.

Related tasks:
- [Optimizing queries with IN predicates (Db2 Performance)]

LIKE_BLANK_INSIGNIFICANT in macro DSN6SPRM

LIKE_BLANK_INSIGNIFICANT subsystem parameter specifies whether blanks are significant when applying the LIKE predicate to a string. If set, the blank insignificant behavior applies.

| Acceptable values: | NO, YES |
| Default:           | NO      |
| DSNZPxxx:          | DSN6SPRM.LIKE_BLANK_INSIGNIFICANT |

NO  LIKE treats trailing blanks within fixed-length character strings as significant.

YES  When the LIKE predicate is applied against fixed-length character column data, Db2 strips trailing blanks from the data before performing the comparison. If the data is all blank, Db2 reduces it to a single blank before performing the comparison.

Related reference:
- [LIKE predicate (Db2 SQL)]

MATERIALIZE_NODET_SQLTUDF in macro DSN6SPRM

Specifies whether the Db2 database manager materializes the result of a user-defined SQL table function that is defined as NOT DETERMINISTIC.

| Acceptable values: | NO, YES |
| Default:           | YES    |
| DSNZPxxx:          | DSN6SPRM MATERIALIZE_NODET_SQLTUDF |

NO  Valid values are YES and NO. The Db2 database manager enables the merging of a user-defined SQL table function that is defined as NOT DETERMINISTIC.

YES  The Db2 database manager materializes the result of a user-defined SQL table function that is defined as NOT DETERMINISTIC. Use this setting to avoid errors in processing SQL statements that reference those user-defined SQL table functions.

Related reference:
- [CREATE FUNCTION (SQL table) (Db2 SQL)]

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST  383
MAX_CONCURRENT_PKG_OPS in macro DSN6SPRM

Specifies the maximum number of automatic bind requests, remote bind requests, remote rebind requests, CREATE TRIGGER statements, and CREATE PROCEDURE statements for packages that can be processed simultaneously.

If the value of the ABIND subsystem parameter is set to NO, the value of MAX_CONCURRENT_PKG_OPS has no effect.

Acceptable values: 1 - 20
Default: 10
DSNZPxxx: DSN6SPRM MAX_CONCURRENT_PKG_OPS

Security parameter: No

1 - 20
The maximum number of automatic bind requests, remote bind requests, remote rebind requests, CREATE TRIGGER statements, and CREATE PROCEDURE statements for packages that can be processed simultaneously.

Related concepts:
- Automatic rebinds (Db2 Application programming and SQL)

Related reference:
- AUTO BIND field (ABIND subsystem parameter)

MOVE_TO_ARCHIVE_DEFAULT in macro DSN6SPRM

Specifies the default for the SYSIBMADM.MOVE_TO_ARCHIVE built-in global variable.

Acceptable values: N, E, Y
Default: N
DSNZPxxx: DSN6SPRM MOVE_TO_ARCHIVE_DEFAULT

Security parameter: No

N Specifies that a delete of a row in an archive-enabled table will not result in storing a copy of the deleted row in the associated archive table.

E Specifies that a delete of a row in an archive-enabled table will result in storing a copy of the deleted row in the associated archive table.

Y Specifies that a delete of a row in an archive-enabled table will result in storing a copy of the deleted row in the associated archive table. Additionally, when the global variable is set to 'Y', an insert or update operation that specifies the archive-enabled table as the target of the statement will result in an error.

Related concepts:
- Archive-enabled tables and archive tables (Introduction to Db2 for z/OS)

Related reference:
- Built-in global variables (Db2 SQL)

NPGTHRSH in macro DSN6SPRM

Specifies that Db2 favors matching index access, through the index with the most matching columns, for tables or partitions that have fewer data pages than the specified value.

Db2 compares the NPGTHRSH value to the NPAGESF column in the SYSIBM.SYSTABLES catalog table for table-level comparisons and the NPAGES column of the SYSIBM.SYSTABSTATS table for partition-level comparisons.
Tip: For best results, set NPGTHRSH to a non-zero value so that Db2 uses matching index access for tables with default statistics. For more information, see Favoring index access (Db2 Performance).

Acceptable values: 0, 1 - 2147483647
Default: 0

DSNZPxxx: DSN6SPRM NPGTHRSH

Security parameter: No

Note:
1. For SAP installations, the default value is 10.

0
Db2 always chooses the access path based on cost estimates, regardless of the number of data pages that statistics indicate for the table or partition.

1 - 2147483647
Db2 uses matching index access for queries that access a table or partition that contains fewer data pages than the specified NPGTHRSH value. Db2 uses the index with the most matching columns.

For default statistics (-1), Db2 12 uses the -1 values for the comparison.

Related reference:
- SYSTABLES catalog table (Db2 SQL)
- SYSTABSTATS catalog table (Db2 SQL)

OPT1ROWBLOCKSORT in macro DSN6SPRM
Specifies whether Db2 explicitly blocks sort operations when the OPTIMIZE FOR 1 ROW clause is specified on a query.

Acceptable values: DISABLE, ENABLE
Default: DISABLE

DSNZPxxx: DSN6SPRM OPT1ROWBLOCKSORT

Security parameter: No

DISABLE
when OPTIMIZE FOR 1 ROW is specified, Db2 avoids access paths that involve sorts. If an access path that avoids a sort exists, it is possible, although unlikely, that an access path that involves a sort is chosen instead. This behavior was always used in DB2 9 and earlier releases.

ENABLE
When OPTIMIZE FOR 1 ROW is specified, Db2 chooses access paths that avoid sorts whenever such a path is available. For example, Db2 must choose between two index access paths. One index is very selective and filters two rows, but it requires a sort. The other index does not filter any rows, but it does not require a sort. If the OPT1ROWBLOCKSORT subsystem parameter is set to ENABLE, Db2 chooses the path that does not require a sort.

Related concepts:
- Optimization for large and small result sets (Introduction to Db2 for z/OS)

Related tasks:
- Minimizing the cost of retrieving few rows (Db2 Performance)
- Optimizing retrieval for a small set of rows (Db2 Application programming and SQL)
Related reference:


**PREVENT_NEW_IXCTRL_PART in macro DSN6SPRM**
The PREVENT_NEW_IXCTRL_PART subsystem parameter determines whether Db2 disallows the creation of new index-controlled partitioned tables. This subsystem parameter ensures that new partitioned tables use table-controlled partitioning, which is the preferred partitioning method for non-universal table spaces.

Acceptable values: YES, NO

Default: YES

DSNZPxxx: DSN6SPRM PREVENT_NEW_IXCTRL_PART

**YES**
Specifies that new partitioned table spaces must use table-controlled partitioning. A CREATE INDEX statement must not attempt to create an index-controlled partitioned table.

**NO**
Specifies that new index-controlled partitioned tables can be created.

**Learn more:** For comprehensive background, how-to information, and examples for various paths for converting your deprecated “classic” partitioned (non-UTS) table spaces to partition-by-range table spaces, see the white paper [Conversion from index-controlled partitioning to Universal Table Space (UTS)](https://www.ibm.com/docs/en/db2).

**Related tasks:**

- Converting partitioned (non-UTS) table spaces to partition-by-range universal table spaces (Db2 Administration Guide)
- Converting table spaces to use table-controlled partitioning (Db2 Administration Guide)
- Creating tables partitioned by data value ranges (Db2 Administration Guide)

**PRIVATE_PROTOCOL in macro DSN6FAC**
Private protocol is no longer supported; however, this subsystem parameter controls package EXECUTE authorization behavior related to remote Db2 requester applications.

Acceptable values: NO, AUTH

Default: NO

DSNZPxxx: DSN6FAC PRIVATE_PROTOCOL

Security parameter: No

**NO**
When a remote Db2 requester application accesses the Db2 subsystem through DRDA protocol, package EXECUTE authority must be explicitly granted at this subsystem. The authority must be granted to the primary ID or an associated secondary ID for packages that the requester application executes as part of the plan.

**AUTH**
When a remote Db2 requester application accesses the Db2 subsystem through DRDA protocol, package EXECUTE authority must be explicitly granted to the plan owner at this subsystem. This plan owner must have EXECUTE authority at this subsystem for packages that the requester application executes as part of the plan. The plan owner needs no other privilege to execute the package.
When a remote Db2 requester application accesses the server Db2 subsystem through DRDA protocol, a plan owner is associated with the remote Db2 requester application plan. For packages that are executed as part of this plan by a remote Db2 requester application using DRDA protocol, the PRIVATE_PROTOCOL parameter has the following effects. This parameter is online updatable, and valid values are NO and AUTH.

**PTASKROL in macro DSN6SYSP**
Indicates whether to roll up accounting trace records from a parallel query task into the originating task’s accounting trace.

Acceptable values: YES, NO
Default: YES
DSNZPxxx: DSN6SYSP PTASKROL
Security parameter: No

**YES**
The originating task is to generate an additional accounting trace record with all the roll-up values from parallel tasks.

**NO** Each parallel task produces its own accounting trace record.

**REALSTORAGE_MAX in macro DSN6SPRM**
Specifies the maximum GB of real and auxiliary storage that Db2 can consume.

Acceptable values: NOLIMIT, 1 - 65535.
Default: NOLIMIT
DSNZPxxx: DSN6SPRM REALSTORAGE_MAX
Security parameter: No

**NOLIMIT**
Real and auxiliary storage consumption for Db2 is not bounded.

**1 - 65535**
If the total real and auxiliary storage in GB exceeds the specified limit, Db2 is terminated. In most cases, set this parameter to twice the amount of real and auxiliary storage that the subsystem might reasonably consume.

**Related reference:**
MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter)

**RESTRICT_ALT_COL_FOR_DCC in macro DSN6SPRM**
Specifies whether restrictions apply to ALTER TABLE ALTER COLUMN statements for target tables that are defined with the DATA CAPTURE CHANGES attribute. Valid values are NO and YES.

Acceptable values: NO, YES
Default: NO
DSNZPxxx: DSN6SPRM RESTRICT_ALT_COL_FOR_DCC
Security parameter: No

**NO** Restrictions do not apply to altering column data types or default values of tables that are defined with the DATA CAPTURE CHANGES attribute. For such tables, ALTER TABLE ALTER COLUMN statements with the SET DATA TYPE, SET DEFAULT, or DROP DEFAULT options are allowed.
restrictions apply to altering column data types or default values of tables that are defined with the DATA CAPTURE CHANGES attribute. ALTER TABLE ALTER COLUMN statements that are issued with the SET DATA TYPE, SET DEFAULT, or DROP DEFAULT options for such tables result in SQLCODE -148.

Related tasks:

- [Altering a table to capture changed data (Db2 Administration Guide)]

Related information:

- [148 (Db2 Codes)]

**SECADM1_INPUT_STYLE in macro DSN6SPRM**

Specifies whether the SECADM1 setting is a character string or a hexadecimal string.

Acceptable values: CHAR, HEX

Default: CHAR

DSNZPxxx DSN6SPRM SECADM1_INPUT_STYLE

Security parameter: Yes

**CHAR**

The SECADM1 value is a standard character string. SECADM1 must be a 1 - 8 character authorization ID. If SECADM1_TYPE=ROLE, SECADM1 must be a 1 - 128 character ordinary SQL identifier.

**HEX**

The SECADM1 values is a hexadecimal character string that represents a Unicode-encoded role name. HEX is valid only when SECADM1_TYPE=ROLE. SECADM1 must be an even number of bytes in the range 2 - 256, consisting entirely of 0 - 9 and A - F.

Related reference:

- [SEC ADMIN 1 TYPE field (SECADM1_TYPE subsystem parameter)]
- [SECURITY ADMIN 1 field (SECADM1 subsystem parameter)]

**SECADM2_INPUT_STYLE in macro DSN6SPRM**

Specifies whether the SECADM2 setting is a character string or a hexadecimal string.

Acceptable values: CHAR, HEX

Default: CHAR

DSNZPxxx DSN6SPRM SECADM2_INPUT_STYLE

Security parameter: Yes

**CHAR**

The SECADM2 value is a standard character string. SECADM2 must be a 1 - 8 character authorization ID. If SECADM2_TYPE=ROLE, SECADM2 must be a 1 - 128 character ordinary SQL identifier.

**HEX**

The SECADM2 values is a hexadecimal character string that represents a Unicode-encoded role name. HEX is valid only when SECADM2_TYPE=ROLE. SECADM2 must be an even number of bytes in the range 2 - 256, consisting entirely of 0 - 9 and A - F.

Related reference:
SECURITY ADMIN 2 field (SECADM2 subsystem parameter)
SEC ADMIN 2 TYPE field (SECADM2_TYPE subsystem parameter)

SIMULATED_CPU_COUNT in macro DSN6SPRM
Specifies the number of CPUs that are being simulated. This parameter is to be used for Db2 optimization modeling only.

Acceptable values: OFF, 1 - 255
Default: OFF
DSNZPxxx: DSN6SPRM SIMULATED_CPU_COUNT
Security parameter: No

OFF
Db2 optimizes for the number of CPUs that it runs on.

1 - 255
Specifies the simulated number of CPUs.

Related tasks:
[Modeling a production environment on a test subsystem (Db2 Performance)]

SIMULATED_CPU_SPEED in macro DSN6SPRM
Specifies the microseconds of execution time for tasks or service request blocks (SRBs) per service unit for the CPU that is being simulated. This parameter is used only for Db2 optimization modeling.

Acceptable values: OFF, 1 - 2147483647
Default: OFF
Online changeable: YES
DSNZPxxx: DSN6SPRM.SIMULATED_CPU_SPEED

OFF
Db2 optimizes for the CPU that it runs on. Use this setting in production environments.

1 - 2147483647
Microseconds of execution time for tasks or service request blocks (SRBs) per service unit for the CPU that is being simulated. If the specified value is 1 -27, Db2 uses a value of 27

Related tasks:
[Modeling a production environment on a test subsystem (Db2 Performance)]

SJTABLES in macro DSN6SPRM
Specifies whether Db2 enables star join processing.

Acceptable values: 0 - 32767
Default: 10
DSNZPxxx: DSN6SPRM SJTABLES
Security parameter: No

0
Star join processing is considered for a query block that has 10 or more tables.

1 - 3
Star join processing is always considered.
Star join processing is considered for a query block that has at least the specified number of tables.

Star join processing is disabled.

Related concepts:
- Star schema access (Db2 Performance)

Related reference:
- STAR JOIN QUERIES field (STARJOIN subsystem parameter)

SMF89 in macro DSN6SYSP
Specifies whether Db2 is to do detailed tracking for measured usage pricing. If you select YES, Db2 does detailed measured usage tracking if SMF type 89 records are activated.

Acceptable values: NO, YES
Default: NO
DSNZPxxx: DSN6SYSP SMF89
Security parameter: No

NO
Db2 does not do detailed measured usage tracking. If the SMF type 89 record is activated, only high-level tracking is recorded in the SMF type 89 record. Selecting NO reduces CPU usage, but also increases the amount of time spent in Db2 as measured by SMF 89.

YES
Db2 invokes a z/OS service on every entry into or exit out of Db2 to ensure accurate tracking.

Recommendation: Select SMF89 YES only if you use measured usage pricing.

SUBQ_MIDX in macro DSN6SPRM
Specifies whether to enable or disable multiple index access on some non-Boolean uncorrelated subquery predicates.

Acceptable values: ENABLE, DISABLE
Default: ENABLE
DSNZPxxx: DSN6SPRM SUBQ_MIDX
Security parameter: No

ENABLE
Multiple index access is allowed for queries that have subquery predicates.

DISABLE
Multiple index access is not allowed for queries that have subquery predicates.

The following example query contains such non-Boolean uncorrelated subquery predicates:

```
SELECT *
FROM T1
WHERE (T1.C1 = ? )
  OR (T1.C2 >= T1.C3
    AND T1.C4 = (SELECT T2.C1
               FROM T2
               WHERE T2.C2 = ?))
```
Related concepts:

- Boolean term predicates (Db2 Performance)
- Multiple index access (ACCESSTYPE='M', 'MX', 'MI', 'MU', 'DX', 'DI', or 'DU') (Db2 Performance)

SUPPRESS_HINT_SQLCODE_DYN in macro DSN6SPRM
Specifies whether Db2 suppresses SQLCODE +394 and SQLCODE +395 when specified access paths are applied for dynamic SQL statements.

| Acceptable values: | NO, STMT, ALL |
| Default: | NO |
| DSNZPxxx: | DSN6SPRM SUPPRESS_HINT_SQLCODE_DYN |
| Security parameter: | No |

**NO**  Db2 issues +394 and +395 for statement-level access paths and PLAN_TABLE access paths.

**STMT**  Db2 suppresses +394 and +395 for statement-level access paths for dynamic SQL statements.

**ALL**  Db2 suppresses +394 and +395 for statement-level access paths and PLAN_TABLE access paths for dynamic SQL statements.

The SUPPRESS_HINT_SQLCODE_DYN subsystem parameter does not apply to static SQL statements or dynamic EXPLAIN statements.

Related tasks:

- Specifying access paths at the statement level (Db2 Performance)
- Specifying access paths in a PLAN_TABLE instance (Db2 Performance)

Related information:

- +394 (Db2 Codes)
- +395 (Db2 Codes)

UNION_COLNAME_7 in macro DSN6SPRM
Specifies the behavior for result column names in UNION queries.

| Acceptable values: | NO, YES |
| Default: | NO |
| DSNZPxxx: | DSN6SPRM UNION_COLNAME_7 |
| Security parameter: | No |

**NO**  If the column name is the same across all sub-queries in the UNION query, the result column name is that column name. Otherwise, the result column is unnamed. DB2 Version 8 introduced this behavior.

**YES**  The column name that is returned in the SQLNAME field of an SQLDA following a DESCRIBE statement, where the result table that is described is the result of a union, is the column name from the first subquery of the union operation. Db2 used this behavior in DB2 Version 7 and earlier.

Related reference:

- DESCRIBE (Db2 SQL)
The included SQLDA (Db2 SQL)

Related information:

**XML_RESTRICT_EMPTY_TAG in DSN6SPRM**

Specifies whether Db2 always serializes an empty XML element using a start-element tag followed by an end-element tag.

- **Acceptable values:** NO, YES
- **Default:** NO
- **DSNZPxxx:** DSN6SPRM XML_RESTRICT_EMPTY_TAG
- **Security parameter:** No

**NO** Db2 serializes an empty XML element in either of the following ways:
  - As a start-element tag followed by an end-element tag. For example:
    `<emptyElement></emptyElement>`
  - As an empty-element tag. For example:
    `<emptyElement/>

**YES**

Db2 always serializes an empty XML element as a start-element tag followed by an end-element tag. For example:

`<emptyElement></emptyElement>`

**Related concepts:**

**ZOSMETRICS in macro DSN6SPRM**

Enables Db2 to gather z/OS metrics, including CPU usage and storage metrics, by using the z/OS Resource Measurement Facility™ (RMF™) interface.

- **Acceptable values:** NO, YES
- **Default:** NO
- **DSNZPxxx:** DSN6SPRM ZOSMETRICS
- **Security parameter:** No

**NO** Db2 does not use RMF to gather z/OS metrics.

**YES**

Db2 uses RMF to gather z/OS metrics, including CPU usage and storage.

The following actions are required to make the RMF interface operational:
  - Verify that the DSNWVRMF module is placed in the LINKLIST library. After installation, refresh the LINKLIST by issuing the following MVS command:
    `f llar,refresh`
  - Start the RMF and RMF Monitor III Sysplex Data Retrieval Service.

In some cases, the z/OS metrics might not be provided, or they might be only partially provided in the Db2 trace. This lack of metrics is likely caused by setup problems that are related to RMF. If support is required, provide the incomplete trace data.
Job DSNTIJUZ: define the Db2 data-only subsystem parameter module

Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

You run job DSNTIJUZ as part of the following Db2 installation, migration tasks, and Db2 12 function-level activation tasks:

- Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL
- Defining Db2 initialization parameters for enabling data sharing
- Defining Db2 initialization parameters for new data sharing members
- Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM
- Defining Db2 12 initialization parameters for migrating data sharing members
- Updating subsystem parameter and application default values
- Updating Db2 initialization parameters for function level activation (Db2 for z/OS What's New?)

The subsystem parameter module consists of the expansion of the following macros that contain the Db2 run-time parameters that you selected by using the ISPF panels:

- DSN6ARVP
- DSN6FAC
- DSN6GRP
- DSN6LOGP
- DSN6SPRM
- DSN6SYSP

You might also be directed by IBM to run it when applying service to Db2. In Db2 data sharing, each member of the group must have its own subsystem parameter module and therefore its own DSNTIJUZ job.

DSNTIJUZ also contains a disabled job step, DSNTIMQ, that can be enabled to set up GENASM entries in SMP/E for the Db2 system parameter (DSN6xxx) macros. The GENASM entries cause service to any of these macros to trigger SMP/E to initiate a reassembly of the Db2 system parameter module. Advanced SMP/E skills are recommended for using GENASM entries. See the warnings in the comment block for job step DSNTIMQ before enabling it.

Adjustments to job DSNTIJUZ

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This
action ensures that members DSNFSYSP, DSNJARVP, DSNJLOGP, DSNTSPRM, DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```
SET BOUNDARY (GLOBAL).
LIST DDDEF (SMPTLIB).
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

Related concepts:

- Application compatibility levels in Db2 (Db2 Application programming and SQL)

Related reference:

- APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter)

Job DSNTIJUA: define data-only application defaults module

Job DSNTIJUA defines the Db2 data-only application defaults module.

Run this job when installing Db2 or migrating to Db2 12, and optionally to change the default SQL processing level (SQLLEVEL) for the precompiler/coprocessor after activating a new Db2 function level.

You run job DSNTIJUA as part of the following tasks:

- Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL
- Defining Db2 initialization parameters for enabling data sharing
- Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM
- Updating subsystem parameter and application default values
- Updating Db2 initialization parameters for function level activation (Db2 for z/OS What's New?)

You might also be directed by IBM to run it when applying service to Db2.

In Db2 data sharing, each member of the group can use a single application defaults module or have its own. The Db2 installation and migration process assumes one module for the whole group and provides a DSNTIJUA job only for the first member.

The default SQL processing level (SQLLEVEL) for the precompiler/coprocessor can be changed anytime, such as after activating a higher Db2 function level. For more information, see PRECOMPILED SQL LEVEL field (SQLLEVEL DECP value)

**Important:** With the exception of SQLLEVEL, updates to application defaults parameters are generally not recommended. In general, you must restart Db2 for changes to any application programming parameters to take effect.

DSNTIJUA also contains a disabled job step, DSNTIMQ, that can be enabled to set up a GENASM entry in SMP/E for the Db2 application programming (DSNHDECM) macro. The GENASM entry causes service to DSNHDECM to trigger SMP/E to initiate a reassembly of the Db2 application defaults module. Advanced SMP/E skills are recommended for using GENASM entries. See the warnings in the comment block for job step DSNTIMQ before enabling it.
Adjustments to job DSNTIJUA

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUA to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUA so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZQ. This action ensures that member DSNARIB is linked with the he application defaults module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```plaintext
SET BOUNDARY (GLOBAL).
LIST DDDEF (SMPTLIB).
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

Related concepts:
- Application programming default values (Introduction to Db2 for z/OS)
- Application defaults parameters

Job DSNTIJUM: define Db2 data-only offline message generator CCSID (DSNHMCID) module

Job DSNTIJUM defines the Db2 data-only offline message generator CCSID (DSNHMCID) module which is needed for message conversion by Db2 applications and utilities.

You run job DSNTIJUM as part of the following tasks Db2 12 installation and migration tasks:

- Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL
- Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM
- Defining Db2 initialization parameters for enabling data sharing

DSNHMCID is link-edited into both the prefix.SDSNEXIT and prefix.SDSNLOAD libraries.

Run this job when installing or migrating to Db2 12.

DSNTIJUM also contains a disabled job step, DSNTIMQ, that can be enabled to set up a GENASM entry in SMP/E for the offline message generator CCSID (DSNHMCIM) macro. The GENASM entry causes service to DSNHMCIM to trigger SMP/E to initiate a reassembly of the DB2 offline message generator CCSID module. Advanced SMP/E skills are recommended for using GENASM entries. See the warnings in the comment block for job step DSNTIMQ before enabling it.

Related tasks:
- Specifying subsystem CCSIDs (Db2 Internationalization Guide)
Job DSNTIJUL: update DDF related BSDS information

Job DSNTIJUL runs the Db2 change log inventory (DSNJU003) to update DDF-related information in the BSDS.

You run job DSNTIJUL as part of the following Db2 12 installation tasks:

- Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL
- Defining Db2 initialization parameters for enabling data sharing
- Updating subsystem parameter and application default values

In Db2 data sharing, you need to run it once for each member of the group. Also, the DDF information must be recorded into the BSDS for each member of the group and therefore each member must have its own DSNTIJUL job.

For updates, the Db2 installation CLIST generates job DSNTIJUL only for changes to any of the following parameters:

- **DB2 LOCATION NAME field**
- **DB2 VTAM LUNAME field**
- **DB2 GENERIC LUNAME field**
- **DB2 VTAM PASSWORD field**
- **DB2 TCP/IP IPNAME field**
- **TCP/IP PORT field**
- **SECURE PORT field**
- **RESYNC PORT field**

Related concepts:

- Bootstrap data set (Introduction to Db2 for z/OS)

Related reference:

- DSNJU003 (change log inventory) (Db2 Utilities)

Directory of subsystem parameters, panel fields, and application default values

Subsystem, installation panel fields, and application default values are settings that apply to an entire Db2 subsystem. Most but not all subsystem parameters correspond to installation panel fields. Also, some panel fields do not correspond to a subsystem parameter.

The following table lists subsystem parameters and macro names, installation panel fields and panel names, and whether each parameter can be updated online. “Yes” in the Update online column means that the parameter can be updated online.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTID cx entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEXP</td>
<td>EXPLAIN PROCESSING</td>
<td>DSN6SPRM</td>
<td>DSNTIPO</td>
<td>Yes</td>
<td>ABEXP</td>
</tr>
<tr>
<td>ABIND</td>
<td>AUTO BIND</td>
<td>DSN6SPRM</td>
<td>DSNTIPO</td>
<td>Yes</td>
<td>ABIND</td>
</tr>
<tr>
<td>ACCEL</td>
<td>ACCELERATION STARTUP</td>
<td>DSN6SPRM</td>
<td>DSNTIPS2</td>
<td>Yes</td>
<td>ACCEL</td>
</tr>
<tr>
<td>ACCELMODEL</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>ACCELMODEL</td>
</tr>
</tbody>
</table>
### Table 42. Directory of subsystem parameters and application default values (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTIDxx entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS_CNTL_MODULE</td>
<td>ACCESS CONTROL</td>
<td>DSN6SYSP</td>
<td>DSN6TP03</td>
<td>Yes</td>
<td>ACCTLMOD</td>
</tr>
<tr>
<td>ACCUMACCC</td>
<td>DDF/RRSAF ACCUM</td>
<td>DSN6SYSP</td>
<td>DSN6TPN</td>
<td>Yes</td>
<td>ACCUMAC</td>
</tr>
<tr>
<td>ACCUMUID</td>
<td>AGGREGATION FIELDS</td>
<td>DSN6SYSP</td>
<td>DSN6TPN</td>
<td>Yes</td>
<td>ACCUMUID</td>
</tr>
<tr>
<td>ADMTPROC</td>
<td>ADMIN SCHEDULER</td>
<td>DSN6SPRM</td>
<td>DSN6TPX</td>
<td>Yes</td>
<td>ADMTPROC</td>
</tr>
<tr>
<td>AEXITLIM</td>
<td>AUTH EXIT LIMIT</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>AEXITLIM</td>
</tr>
<tr>
<td>AGCCSID</td>
<td>ASCII CCSID</td>
<td>DSN6HDECp</td>
<td>DSN6TPF</td>
<td>Yes</td>
<td>AGCCSID</td>
</tr>
<tr>
<td>—</td>
<td>dbname ALL</td>
<td>DSN6SPRM</td>
<td>DSN6TPS</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ALTERNATE_CP</td>
<td>ALTERNATE COPY POOL</td>
<td>DSN6SPRM</td>
<td>DSN6TP62</td>
<td>Yes</td>
<td>BSACP</td>
</tr>
<tr>
<td>AMCCSID</td>
<td>ASCII CCSID</td>
<td>DSN6HDECp</td>
<td>DSN6TPF</td>
<td>Yes</td>
<td>AMCCSID</td>
</tr>
<tr>
<td>APPENSCH</td>
<td>APPLICATION ENCODING</td>
<td>DSN6HDECp</td>
<td>DSN6TPF</td>
<td>Yes</td>
<td>APPENSCH</td>
</tr>
<tr>
<td>APPLCOMPAT</td>
<td>APPL COMPAT LEVEL</td>
<td>DSN6SPRM</td>
<td>DSN6TP00</td>
<td>Yes</td>
<td>APPLCOMPAT</td>
</tr>
<tr>
<td>ARC2FRST1</td>
<td>Archive logs: COPY 1 PREFIX</td>
<td>DSN6ARVP</td>
<td>DSN6TPH</td>
<td>Yes</td>
<td>ARCHPRE1</td>
</tr>
<tr>
<td>ARC2FRST2</td>
<td>Archive logs: COPY 2 PREFIX</td>
<td>DSN6ARVP</td>
<td>DSN6TPH</td>
<td>Yes</td>
<td>ARCHPRE2</td>
</tr>
<tr>
<td>ARCRETN</td>
<td>RETENTION PERIOD</td>
<td>DSN6ARVP</td>
<td>DSN6TPA</td>
<td>Yes</td>
<td>ARCRETN</td>
</tr>
<tr>
<td>ARCHWRTC</td>
<td>WTOR ROUTE CODE</td>
<td>DSN6ARVP</td>
<td>DSN6TPA</td>
<td>Yes</td>
<td>ARCHWRTC</td>
</tr>
<tr>
<td>ARCWCTR</td>
<td>WRITE TO OPER</td>
<td>DSN6ARVP</td>
<td>DSN6TPA</td>
<td>Yes</td>
<td>ARCWCTR</td>
</tr>
<tr>
<td>ARCT2FRST</td>
<td>READ COPY2 ARCHIVE</td>
<td>DSN6LOGP</td>
<td>DSN6TOPO</td>
<td>Yes</td>
<td>ARCT2FRST</td>
</tr>
<tr>
<td>ASCSID</td>
<td>ASCII CCSID</td>
<td>DSN6HDECp</td>
<td>DSN6TPF</td>
<td>Yes</td>
<td>ASCSIDE</td>
</tr>
<tr>
<td>AUDITTS1</td>
<td>AUDIT TRACE</td>
<td>DSN6SYSP</td>
<td>DSN6TPN</td>
<td>Yes</td>
<td>AUDIT</td>
</tr>
<tr>
<td>AUTH</td>
<td>USE PROTECTION</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>PROAUTH</td>
</tr>
<tr>
<td>AUTHCACHE</td>
<td>PLAN AUTH CACHE</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>AUTHCACHE</td>
</tr>
<tr>
<td>AUTHEXIT_CACHEREFRESH</td>
<td>AUTH EXIT CACHE REFRESH</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>AECR</td>
</tr>
<tr>
<td>AUTH_COMPATIBILITY</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>DSN6TPM</td>
<td>Yes</td>
<td>AUTCMPAT</td>
</tr>
<tr>
<td>AUTHEXIT_CHECK</td>
<td>AUTH EXIT CHECK</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>RACK</td>
</tr>
<tr>
<td>BACKDUR</td>
<td>BACKOUT DURATION</td>
<td>DSN6SYSP</td>
<td>DSN6TLPL1</td>
<td>Yes</td>
<td>BACKDUR</td>
</tr>
<tr>
<td>BIF_COMPATIBILITY</td>
<td>BIF COMPATIBILITY</td>
<td>DSN6SPRM</td>
<td>DSN6TPX</td>
<td>Yes</td>
<td>BIFCP</td>
</tr>
<tr>
<td>BINDNV</td>
<td>BIND NEW PACKAGE</td>
<td>DSN6SPRM</td>
<td>DSN6TP1P</td>
<td>Yes</td>
<td>BINDNV</td>
</tr>
<tr>
<td>ELKSIZE</td>
<td>BLOCK SIZE</td>
<td>DSN6ARVP</td>
<td>DSN6TPA</td>
<td>Yes</td>
<td>ELKSIZE</td>
</tr>
<tr>
<td>BMPOUT</td>
<td>IMS BMP TIMEOUT</td>
<td>DSN6SPRM</td>
<td>DSN6TP1</td>
<td>Yes</td>
<td>BMTOUT</td>
</tr>
<tr>
<td>CACHEDYN</td>
<td>CACHE DYNAMIC SQL</td>
<td>DSN6SPRM</td>
<td>DSN6TP8</td>
<td>Yes</td>
<td>CACHEDUM</td>
</tr>
<tr>
<td>CACHEDYN_STABILIZATION</td>
<td>CACHE DYN STABILITY</td>
<td>DSN6SPRM</td>
<td>DSN6TP8</td>
<td>Yes</td>
<td>CDST</td>
</tr>
<tr>
<td>CACHEPAC</td>
<td>PACKAGE AUTH CACHE</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>CACHEPAC</td>
</tr>
<tr>
<td>CACHERAC</td>
<td>ROUTINE AUTH CACHE</td>
<td>DSN6SPRM</td>
<td>DSN6TPP</td>
<td>Yes</td>
<td>CACHERAC</td>
</tr>
<tr>
<td>CATALOG</td>
<td>CATALOG ALIAS</td>
<td>DSN6SPRM</td>
<td>DSN6TPA2</td>
<td>Yes</td>
<td>VCATALOG</td>
</tr>
<tr>
<td>CATDDACL</td>
<td>DIRECTORY AND CATALOG DATA</td>
<td>DSN6SPRM</td>
<td>DSN6TPA2</td>
<td>Yes</td>
<td>CATDDACL</td>
</tr>
<tr>
<td>CATDMGCL</td>
<td>DIRECTORY AND CATALOG DATA</td>
<td>DSN6SPRM</td>
<td>DSN6TPA2</td>
<td>Yes</td>
<td>CATDMGCL</td>
</tr>
<tr>
<td>CATDSTCL</td>
<td>DIRECTORY AND CATALOG DATA</td>
<td>DSN6SPRM</td>
<td>DSN6TPA2</td>
<td>Yes</td>
<td>CATDSTCL</td>
</tr>
<tr>
<td>CATXDAACL</td>
<td>DIRECTORY AND CATALOG INDEXES</td>
<td>DSN6SPRM</td>
<td>DSN6TPA2</td>
<td>Yes</td>
<td>CATXDAACL</td>
</tr>
</tbody>
</table>

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTID entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATXMGC1</td>
<td>DIRECTORY AND CATALOG INDEXES</td>
<td>DSN6SPRM</td>
<td>DSNTPA2</td>
<td>Yes</td>
<td>CATXMGC1</td>
</tr>
<tr>
<td>CATXSTCL</td>
<td>DIRECTORY AND CATALOG INDEXES</td>
<td>DSN6SPRM</td>
<td>DSNTPA2</td>
<td>Yes</td>
<td>CATXSTCL</td>
</tr>
<tr>
<td>CDDS_MODE</td>
<td>—</td>
<td>DSN6LOGP</td>
<td>—</td>
<td>Yes</td>
<td>CDDS_MODE</td>
</tr>
<tr>
<td>CDDS_PREFIX</td>
<td>—</td>
<td>DSN6LOGP</td>
<td>—</td>
<td>Yes</td>
<td>CDDS_MODE</td>
</tr>
<tr>
<td>CDSSRDEF</td>
<td>CURRENT DEGREE</td>
<td>DSN6SPRM</td>
<td>DSNTP11</td>
<td>Yes</td>
<td>CDSSRDEF</td>
</tr>
<tr>
<td>CHECK_FASTREPLICATION</td>
<td>FAST REPLICATION</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>CHKFASRP</td>
</tr>
<tr>
<td>CHGDC</td>
<td>DPROP SUPPORT</td>
<td>DSN6SPRM</td>
<td>DSNTP0</td>
<td>Yes</td>
<td>CHGDC</td>
</tr>
<tr>
<td>CHKFREQ</td>
<td>CHECKPOINT FREQ</td>
<td>DSN6SYSYP</td>
<td>DSNTP111</td>
<td>Yes</td>
<td>CHKFREQ</td>
</tr>
<tr>
<td>CHKLOGR</td>
<td>RECORDS/CHECKPOINT</td>
<td>DSN6SYSYP</td>
<td>DSNTP111</td>
<td>Yes</td>
<td>CHKLOGR</td>
</tr>
<tr>
<td>CHKMINS</td>
<td>MINUTES/CHECKPOINT</td>
<td>DSN6SYSYP</td>
<td>DSNTP111</td>
<td>Yes</td>
<td>CKPTTIME</td>
</tr>
<tr>
<td>CHKTYPE</td>
<td>CHECKPOINT TYPE</td>
<td>DSN6SYSYP</td>
<td>DSNTP111</td>
<td>Yes</td>
<td>CHKTYPE</td>
</tr>
<tr>
<td>CMSSTAT</td>
<td>DDF THREADS</td>
<td>DSN6FAC</td>
<td>DSNTP13</td>
<td>Yes</td>
<td>CMSSTAT</td>
</tr>
<tr>
<td>COMCRT</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>COMCRT</td>
</tr>
<tr>
<td>COMPACT</td>
<td>COMPACT DATA</td>
<td>DSN6ARVP</td>
<td>DSNTPA</td>
<td>Yes</td>
<td>COMPACT</td>
</tr>
<tr>
<td>COMPRESS_DIRLOB</td>
<td>COMPRESS DB2 DIR LOBS</td>
<td>DSN6SPRM</td>
<td>DSNTPA2</td>
<td>Yes</td>
<td>COMPRESS_DIRLOB</td>
</tr>
<tr>
<td>COMPRESS_SPT01</td>
<td>COMPRESS SPT01</td>
<td>DSN6SPRM</td>
<td>DSNTPA2</td>
<td>Yes</td>
<td>COMPRESS_SPT01</td>
</tr>
<tr>
<td>CONDBAT</td>
<td>MAX REMOTE CONNECTED</td>
<td>DSN6SYSYP</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>CONDBAT</td>
</tr>
<tr>
<td>COPY_FASTREPLICATION</td>
<td>COPY FAST REPLICATION</td>
<td>DSN6SPRM</td>
<td>DSNTP16</td>
<td>Yes</td>
<td>COPY_A</td>
</tr>
<tr>
<td>CTTHREAD</td>
<td>MAX USERS</td>
<td>DSN6SYSYP</td>
<td>DSNTP11</td>
<td>Yes</td>
<td>DBACRWW</td>
</tr>
<tr>
<td>DBACRWW</td>
<td>DBADM CREATE AUTH</td>
<td>DSN6SPRM</td>
<td>DSNTP111</td>
<td>Yes</td>
<td>DBACRWW</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE FORMAT</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DATE</td>
</tr>
<tr>
<td>DATELEN</td>
<td>LOCAL DATE LENGTH</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DATELEN</td>
</tr>
<tr>
<td>DB2SUPLD</td>
<td>—</td>
<td>DSNHDECP</td>
<td>—</td>
<td>Yes</td>
<td>DB2SUPLD</td>
</tr>
<tr>
<td>DDF</td>
<td>DDF STARTUP OPTION</td>
<td>DSN6FAC</td>
<td>DSNTP1R</td>
<td>Yes</td>
<td>DDF</td>
</tr>
<tr>
<td>DDF_COMPATIBILITY</td>
<td>—</td>
<td>DSN6FAC</td>
<td>—</td>
<td>Yes</td>
<td>DDF_CMPAT</td>
</tr>
<tr>
<td>DDL_MATERIALIZATION</td>
<td>DDL MATERIALIZATION</td>
<td>DSN6SPRM</td>
<td>DSNTP11</td>
<td>Yes</td>
<td>DDLM</td>
</tr>
<tr>
<td>DDLTOX</td>
<td>DATA DEF TIMEOUT</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>DDLTOX</td>
</tr>
<tr>
<td>DEALLCT</td>
<td>DEALLOC PERIOD</td>
<td>DSN6LOGP</td>
<td>DSNTPA</td>
<td>Yes</td>
<td>ARCHDEAP</td>
</tr>
<tr>
<td>DECARTH</td>
<td>DECIMAL ARITHMETIC</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DECARTH</td>
</tr>
<tr>
<td>DECDIV3</td>
<td>MINIMUM DIVIDE SCALE</td>
<td>DSN6SPRM</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DECDIV3</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>DECIMAL POINT IS</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DECIMAL</td>
</tr>
<tr>
<td>DEF_DECFLOAT_ROUND_MODE</td>
<td>DECFLOAT ROUNDING MODE</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEFDFPRM</td>
</tr>
<tr>
<td>DEFAULT_INSERT_ALGORITHM</td>
<td>DEFAULT INSERT ALGORITHM</td>
<td>DSN6SPRM</td>
<td>DSNTP11</td>
<td>Yes</td>
<td>DINA</td>
</tr>
<tr>
<td>DEFLANG</td>
<td>LANGUAGE DEFAULT</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEFLANG</td>
</tr>
<tr>
<td>DEFTLD</td>
<td>UNKNOWN AUTHID</td>
<td>DSN6SPRM</td>
<td>DSNTP11</td>
<td>Yes</td>
<td>PROTUNKN</td>
</tr>
<tr>
<td>DEL_CFSTRUCTS_ON_RESTART</td>
<td>DEF CF STRUCTS</td>
<td>DSN6SYSYP</td>
<td>DSNTPK</td>
<td>Yes</td>
<td>DEF CF STRUCTS</td>
</tr>
<tr>
<td>DELIM</td>
<td>STRING DELIMITER</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEFSTRNG</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Field name</td>
<td>Macro</td>
<td>Panel</td>
<td>Update online</td>
<td>DSNTDIＹ(entry or z/OSMF variable)</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------</td>
<td>-------------</td>
<td>--------</td>
<td>---------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>DESCSTAT</td>
<td>DESCRIBE FOR STATIC</td>
<td>DSN6SPRM</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DECSTAT</td>
</tr>
<tr>
<td>DISABLE_EDMRTS</td>
<td>DISABLE EDM RTS</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes</td>
<td>DEDR</td>
</tr>
<tr>
<td>DISALLOW_SEL INTO UNION</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>DLDFREQ</td>
<td>LEVELID UPDATE FREQ</td>
<td>DSN6SYS</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>DLDFREQ</td>
</tr>
<tr>
<td>DLTOUT</td>
<td>DL/I BATCH TIMEOUT</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>DLTOUT</td>
</tr>
<tr>
<td>DPSEGSZ</td>
<td>DEFAULT PARTITION SEG SIZE</td>
<td>DSN6SYS</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>DPSEGSZ</td>
</tr>
<tr>
<td>DSKEEP</td>
<td>DATA SHARING</td>
<td>DSN6GRP</td>
<td>DSNTP4</td>
<td>DSKEEP</td>
<td>DSKEEP</td>
</tr>
<tr>
<td>DSMAX</td>
<td>DSN6SPRM</td>
<td>DSNTPC</td>
<td></td>
<td>Yes</td>
<td>DSMAX</td>
</tr>
<tr>
<td>DSQL2DEL</td>
<td>DIST SQL STR DELIMTR</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEDPDS</td>
</tr>
<tr>
<td>DSTIME</td>
<td>DATASET STATS TIME</td>
<td>DSN6SYS</td>
<td>DSNTPN</td>
<td>Yes</td>
<td>DSTIME</td>
</tr>
<tr>
<td>DSVCI</td>
<td>VARY DS CONTROL INTERVAL</td>
<td>DSN6SYS</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>DSVCI</td>
</tr>
<tr>
<td>DYNRULS</td>
<td>USE FOR DYNAMICRULES</td>
<td>DSNHDECP</td>
<td>DSNTP1</td>
<td>DYNRULS</td>
<td>DYNRULS</td>
</tr>
<tr>
<td>EDM_SKELETON_POOL</td>
<td>EDM SKELETON POOL SIZE</td>
<td>DSN6SPRM</td>
<td>DSNTPC</td>
<td>Yes</td>
<td>EDMSKP</td>
</tr>
<tr>
<td>EDMDBDC</td>
<td>EDM DB CACHE</td>
<td>DSN6SPRM</td>
<td>DSNTPC</td>
<td>Yes</td>
<td>EDMDBDC</td>
</tr>
<tr>
<td>EDMSTMTC</td>
<td>EDM STATEMENT CACHE</td>
<td>DSN6SPRM</td>
<td>DSNTPC</td>
<td>Yes</td>
<td>EDMSTMTC</td>
</tr>
<tr>
<td>EDPROP</td>
<td>DPROP SUPPORT</td>
<td>DSN6SPRM</td>
<td>DSNTP0</td>
<td>Yes</td>
<td>EDPROP</td>
</tr>
<tr>
<td>EN_DS</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td>Yes</td>
<td>ENSHCME</td>
</tr>
<tr>
<td>ENCRYPTION_KEYLABEL</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td>Yes (page 412)</td>
<td>ENKL</td>
</tr>
<tr>
<td>ENSCHEME</td>
<td>DEF ENCODING SCHEME</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>ENPJISJ</td>
<td>ENPJISJ</td>
</tr>
<tr>
<td>EVALUNC</td>
<td>EVALUATE UNCOMMITTED</td>
<td>DSN6SPRM</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>EVALUNC</td>
</tr>
<tr>
<td>EXTRAREQ</td>
<td>EXTRA BLOCKS REQ</td>
<td>DSN6SYS</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>EXTRAREQ</td>
</tr>
<tr>
<td>EXTRASRV</td>
<td>EXTRA BLOCKS SRV</td>
<td>DSN6SYS</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>EXTRASRV</td>
</tr>
<tr>
<td>EXTSEC</td>
<td>EXTENDED SECURITY</td>
<td>DSN6SYS</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>EXTSEC</td>
</tr>
<tr>
<td>FCCOPYDDN</td>
<td>DEFAULT TEMPLATE</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCCDTEMP</td>
</tr>
<tr>
<td>FlashCopy_COPY</td>
<td>COPY</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCCP</td>
</tr>
<tr>
<td>FlashCopy_LOAD</td>
<td>LOAD</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCLP</td>
</tr>
<tr>
<td>FlashCopy_PPRC</td>
<td>FlashCopy PPRC</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCPRC</td>
</tr>
<tr>
<td>FlashCopy_Rebuild_Index</td>
<td>REBUILD INDEX</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCROT</td>
</tr>
<tr>
<td>FlashCopy_Reorg_Index</td>
<td>REORG INDEX</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCRBI</td>
</tr>
<tr>
<td>FlashCopy_Reorg_Ts</td>
<td>REORG TABLESPACE</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>FCRBI</td>
</tr>
<tr>
<td>FlashCopy_Xrcp</td>
<td>FlashCopy XRCP</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>GCCSID</td>
<td>EBCDIC CCSID</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>GCCSID</td>
<td>GCCSID</td>
</tr>
<tr>
<td>GET_ACCEL_ARCHIVE</td>
<td>GET ACCEL ARCHIVE</td>
<td>DSN6SPRM</td>
<td>DSNTP2</td>
<td>Yes</td>
<td>CGAA</td>
</tr>
<tr>
<td>GRPNAME</td>
<td>GROUP NAME</td>
<td>DSN6GRP</td>
<td>DSNTPK</td>
<td>GRPNAME</td>
<td>GRPNAME</td>
</tr>
<tr>
<td>HONOR_KEEPDICTIONARY</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td>Yes</td>
<td>HNRKPDIC</td>
</tr>
<tr>
<td>IAUTH_MODULE</td>
<td>IDENTIFY/AUTH</td>
<td>DSN6SYS</td>
<td>DSNTP3</td>
<td>Yes</td>
<td>IAUTHMOD</td>
</tr>
<tr>
<td>DBACK</td>
<td>MAX BATCH CONNECT</td>
<td>DSN6SPM</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>NUMCONB1</td>
</tr>
</tbody>
</table>

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST  399
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTID:EX entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFORE</td>
<td>MAX TSO CONNECT</td>
<td>DSN6SYSP</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>NUMCONTX</td>
</tr>
<tr>
<td>DTHTOIN</td>
<td>IDLE THREAD TIMEOUT</td>
<td>DSN6FAC</td>
<td>DSNTP5</td>
<td>Distributed</td>
<td>data facility panel 2</td>
</tr>
<tr>
<td>IDXBPPOOL</td>
<td>DEFAULT BUFFER POOL FOR USER INDEXES</td>
<td>DSN6SYSP</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>IDXBPPOOL</td>
</tr>
<tr>
<td>IGNORTIN</td>
<td>IGNORE SORTNUM STMT</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>IGNORTIN</td>
</tr>
<tr>
<td>IMMEDWRI</td>
<td>IMMEDIATE WRITE</td>
<td>DSN6GRP</td>
<td>DSNTPS</td>
<td>Yes</td>
<td>IMMEDWRI</td>
</tr>
<tr>
<td>MPDSDEF</td>
<td>DEFINE DATA SETS</td>
<td>DSN6SYSP</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>IMPDSDEF</td>
</tr>
<tr>
<td>XPDSIZE</td>
<td>—</td>
<td>DSN6SYSP</td>
<td>—</td>
<td>Yes</td>
<td>IMPDSIZE</td>
</tr>
<tr>
<td>IMPLICIT_TIMEZONE</td>
<td>IMPLICIT TIME ZONE</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>IMTZ</td>
</tr>
<tr>
<td>MPTKMOD</td>
<td>—</td>
<td>DSN6SYSP</td>
<td>—</td>
<td>Yes</td>
<td>MPTKMOD</td>
</tr>
<tr>
<td>IMPTSCMP</td>
<td>USE DATA COMPRESSION</td>
<td>DSN6SYSP</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>IMPTSCMP</td>
</tr>
<tr>
<td>INDEX_CLEANUP_THREADS</td>
<td>INDEX CLEANUP THREADS</td>
<td>DSN6SPRM</td>
<td>DSNTP3</td>
<td>Yes</td>
<td>IXCU</td>
</tr>
<tr>
<td>INDEX_MEMORY_CONTROL</td>
<td>INDEX MEMORY CONTROL</td>
<td>DSN6SYSP</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>IXMC</td>
</tr>
<tr>
<td>INLISTP</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>INLISTP</td>
</tr>
<tr>
<td>RLMAUT</td>
<td>AUTO START</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>IRLMAUTO</td>
<td></td>
</tr>
<tr>
<td>RLMPROC</td>
<td>PROC NAME</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>IRLMPROC</td>
<td></td>
</tr>
<tr>
<td>RLMRWT</td>
<td>RESOURCE TIMEOUT</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>IRLMWAIT</td>
<td></td>
</tr>
<tr>
<td>RLMSID</td>
<td>SUBSYSTEM NAME</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>IRLMSSID</td>
<td></td>
</tr>
<tr>
<td>RLMSWT</td>
<td>TIME TO AUTOSTART</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>IRLMSTTO</td>
<td></td>
</tr>
<tr>
<td>X_TB_PART_CONV_EXCLUDE</td>
<td>EXCLUDE PART KEY COLUMNS</td>
<td>DSN6SPRM</td>
<td>DSNTP2</td>
<td>Yes</td>
<td>XPKE</td>
</tr>
<tr>
<td>IXQTY</td>
<td>INDEX SPACE ALLOCATION</td>
<td>DSN6SYSP</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>IXQTY</td>
</tr>
<tr>
<td>BACOUT</td>
<td>LIMIT BACKOUT</td>
<td>DSN6SYSP</td>
<td>DSNTP1</td>
<td>LBACKOUT</td>
<td></td>
</tr>
<tr>
<td>LB_TYPE</td>
<td>LOCALE LC_CTYPE</td>
<td>DSNHDECP</td>
<td>DSNTPF</td>
<td>LCCTYPE</td>
<td></td>
</tr>
<tr>
<td>LIKE_BLANK_INSIGNIFICANT</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>LBIN</td>
</tr>
<tr>
<td>LOB_INLINE_LENGTH</td>
<td>LOB INLINE LENGTH</td>
<td>DSN6SYSP</td>
<td>DSNTPD</td>
<td>Yes</td>
<td>LOBINLEN</td>
</tr>
<tr>
<td>LDRTHLD</td>
<td>LONG-RUNNING READER</td>
<td>DSN6SPRM</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>LDRTHLD</td>
</tr>
<tr>
<td>MAINTYPE</td>
<td>CURRENT MAINT TYPES</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes</td>
<td>MAINTYPE</td>
</tr>
<tr>
<td>MATERIALIZE_NODESLTQDF</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>MNSU</td>
</tr>
<tr>
<td>MAXSORT_IN_MEMORY</td>
<td>MAX IN-MEMORY SORT SIZE</td>
<td>DSN6SPRM</td>
<td>DSNTPC</td>
<td>Yes</td>
<td>ARMCXMAX</td>
</tr>
<tr>
<td>MAX_CONCURRENT_PKG_OPS</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>MXCONPKG</td>
</tr>
<tr>
<td>MAX_NUM_CUR</td>
<td>MAX OPEN CURSORS</td>
<td>DSN6SPRM</td>
<td>DSNTPX</td>
<td>Yes</td>
<td>MXXNUMCUR</td>
</tr>
<tr>
<td>MAX_ST_PROC</td>
<td>MAX STORED PROCES</td>
<td>DSN6SPRM</td>
<td>DSNTPX</td>
<td>Yes</td>
<td>MXSTPROC</td>
</tr>
<tr>
<td>MAXARCH</td>
<td>RECORDING MAX</td>
<td>DSN6LOGP</td>
<td>DSNTPA</td>
<td>ARCHMAXV</td>
<td></td>
</tr>
<tr>
<td>MAXCONQN</td>
<td>CONN QUEUE MAX DEPTH</td>
<td>DSN6FAC</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>MAXCONQN</td>
</tr>
<tr>
<td>MAXCONQW</td>
<td>CONN QUEUE MAX WAIT</td>
<td>DSN6FAC</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>MAXCONQW</td>
</tr>
<tr>
<td>MAXDBAIL</td>
<td>MAX REMOTE ACTIVE</td>
<td>DSN6SYSP</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>NUMCONRM</td>
</tr>
<tr>
<td>MAXKEEPD</td>
<td>MAX KEPT DYN STMTS</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>MAXKEEPD</td>
</tr>
</tbody>
</table>
Table 42. Directory of subsystem parameters and application default values (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXOFILR</td>
<td>MAX OPEN FILE REFS</td>
<td>DSN6SYSP</td>
<td>DSNTP1F</td>
<td>Yes MAXOFILR</td>
</tr>
<tr>
<td>MAXBLK</td>
<td>RID POOL SIZE</td>
<td>DSN6SPRM</td>
<td>DSNTP1C</td>
<td>Yes RIDBLK</td>
</tr>
<tr>
<td>MAXRTU</td>
<td>READ TAPE UNITS</td>
<td>DSN6LOGP</td>
<td>DSNTP1A</td>
<td>Yes ARCHMRTU</td>
</tr>
<tr>
<td>MAXTEMPS</td>
<td>MAX TEMP STG/AGENT</td>
<td>DSN6SPRM</td>
<td>DSNTP1H</td>
<td>Yes MAXTEMPS</td>
</tr>
<tr>
<td>MAXTEMPS_RID</td>
<td>MAX TEMP RID</td>
<td>DSN6SPRM</td>
<td>DSNTP1H</td>
<td>Yes MXTRRID</td>
</tr>
<tr>
<td>MAXTYPE1</td>
<td>MAX INACTIVE DBATS</td>
<td>DSN6FAC</td>
<td>DSNTP5</td>
<td>Yes MAXTYPE1</td>
</tr>
<tr>
<td>MCCSID</td>
<td>EBCDIC CCSID</td>
<td>DSNHDECP</td>
<td>DSNTPF</td>
<td>MCCSID</td>
</tr>
<tr>
<td>MEMBNAME</td>
<td>MEMBER NAME</td>
<td>DSN6GRP</td>
<td>DSNTPK</td>
<td>MEMBNAME</td>
</tr>
<tr>
<td>MGEXTSZ</td>
<td>OPTIMIZE EXTENT SIZING</td>
<td>DSN6SYSP</td>
<td>DSNTP7</td>
<td>Yes MGEXTSZ</td>
</tr>
<tr>
<td>MIXED</td>
<td>MIXED DATA</td>
<td>DSNHDECP</td>
<td>DSNTPF</td>
<td>DEF MIXED</td>
</tr>
<tr>
<td>MON</td>
<td>MONITOR TRACE</td>
<td>DSN6SYSP</td>
<td>DSNTPN</td>
<td>MON</td>
</tr>
<tr>
<td>MONSIZE</td>
<td>MONITOR SIZE</td>
<td>DSN6SYSP</td>
<td>DSNTPN</td>
<td>MON</td>
</tr>
<tr>
<td>MOVE_TO_ARCHIVE_DEFAULT</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes MOVE_TO_ARCHIVE_DEFAULT</td>
</tr>
<tr>
<td>MXDTCACH</td>
<td>MAX DATA CACHING</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes MXDTCACH</td>
</tr>
<tr>
<td>NPGTHRSH</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes NPGTHRSH</td>
</tr>
<tr>
<td>NMLKTS</td>
<td>LOCKS PER TABLE(SPACE)</td>
<td>DSN6SPRM</td>
<td>DSNTPJ</td>
<td>Yes NMLKTS</td>
</tr>
<tr>
<td>NUMLKUS</td>
<td>LOCKS PER USER</td>
<td>DSN6SPRM</td>
<td>DSNTPJ</td>
<td>Yes NMLKUS</td>
</tr>
<tr>
<td>OBJECT_CREATE_FORMAT</td>
<td>OBJECT CREATE FORMAT</td>
<td>DSN6SPRM</td>
<td>DSNTP7</td>
<td>Yes OBCF</td>
</tr>
<tr>
<td>OPT1ROWBLOCKSORT</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes OPT1ROWBLOCKSORT</td>
</tr>
<tr>
<td>OPTHINTS</td>
<td>OPTIMIZATION HINTS</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes OPTHINTS</td>
</tr>
<tr>
<td>OTC_LICENSE</td>
<td>OTC LICENSE USAGE</td>
<td>DSN6SYSP</td>
<td>DSNTP2</td>
<td>OTC_LICENSE</td>
</tr>
<tr>
<td>OUTBUFF</td>
<td>OUTPUT BUFFER</td>
<td>DSN6LOGP</td>
<td>DSNTPL</td>
<td>OUTBUFF</td>
</tr>
<tr>
<td>PADIX</td>
<td>PAD INDEXES BY DEFAULT</td>
<td>DSN6SPRM</td>
<td>DSNTP7</td>
<td>Yes PADIX</td>
</tr>
<tr>
<td>PADNSTR</td>
<td>PAD NUL-TERMINATED</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>PADNSTR</td>
</tr>
<tr>
<td>PAGESET_PAGE_NUMBERING</td>
<td>PAGE SET PAGE NUMBERING</td>
<td>DSN6SPRM</td>
<td>DSNTP7</td>
<td>Yes PSPN</td>
</tr>
<tr>
<td>PARA_EFF</td>
<td>PARALLELISM EFFICIENCY</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes PARA_EFF</td>
</tr>
<tr>
<td>PARAMDEG</td>
<td>MAX DEGREE</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes PARAMDEG</td>
</tr>
<tr>
<td>PARAMDEG_DPSI</td>
<td>MAX DEGREE FOR DPSI</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes DEGD</td>
</tr>
<tr>
<td>PARAMDEG_UTIL</td>
<td>MAX UTILS PARALLELISM</td>
<td>DSN6SPRM</td>
<td>DSNTPH</td>
<td>Yes UMD</td>
</tr>
<tr>
<td>PCLOSEN</td>
<td>RO SWITCH CHKPTS</td>
<td>DSN6SYSP</td>
<td>DSNTP1</td>
<td>Yes PCLOSEN</td>
</tr>
<tr>
<td>PCTFREE_UPD</td>
<td>PERCENT FREE FOR UPDATE</td>
<td>DSN6SPRM</td>
<td>DSNTP7</td>
<td>Yes PFUP</td>
</tr>
<tr>
<td>PCLOSET</td>
<td>RO SWITCH TIME</td>
<td>DSN6SYSP</td>
<td>DSNTP1</td>
<td>Yes PKRC</td>
</tr>
<tr>
<td>PEER_RECOVERY</td>
<td>PEER RECOVERY</td>
<td>DSN6GRP</td>
<td>DSNTPK</td>
<td>Yes PEERRECO</td>
</tr>
<tr>
<td>PKGREL_COMMIT</td>
<td>PACKAGE RELEASE COMMIT</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes PKRC</td>
</tr>
<tr>
<td>PLANMgmt SCOPE</td>
<td>PLAN MANAGEMENT</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes PLANMgmt SCOPE</td>
</tr>
<tr>
<td>PLANMgmtSCOPE</td>
<td>PLAN MANAGEMENT SCOPE</td>
<td>DSN6SPRM</td>
<td>DSNTP8</td>
<td>Yes PMSCOPE</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Field name</td>
<td>Macro</td>
<td>Panel</td>
<td>Update online</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------</td>
<td>---------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>POOLINAC</td>
<td>POOL THREAD TIMEOUT</td>
<td>DSN6FAC</td>
<td>DSN1TP</td>
<td>Yes</td>
</tr>
<tr>
<td>PRIQTY</td>
<td>PRIMARY QUANTITY</td>
<td>DSN6ARVP</td>
<td>DSN1PA</td>
<td>Yes</td>
</tr>
<tr>
<td>PREVENTALTERTB_LIMITKEY</td>
<td>PREVENT ALTER LIMITKEY</td>
<td>DSN6SPRM</td>
<td>DSN1TP2</td>
<td>Yes</td>
</tr>
<tr>
<td>PREVENT_NEW_IXCTRL_PART</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>PRIVATE_PROTOCOL</td>
<td>—</td>
<td>DSN6FAC</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>PROFILE_AUTOSTART</td>
<td>PROFILE AUTOSTART</td>
<td>DSN6SYSP</td>
<td>DSN1PO</td>
<td>PFAS</td>
</tr>
<tr>
<td>PROTECT</td>
<td>ARCHIVE LOG RACF</td>
<td>DSN6ARVP</td>
<td>DSN1TP</td>
<td>Yes</td>
</tr>
<tr>
<td>PTA KROL</td>
<td>—</td>
<td>DSN6SYSP</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>QUERY_ACCEL_OPTIONS</td>
<td>ACCELERATION OPTIONS</td>
<td>DSN6SPRM</td>
<td>DSN1PS2</td>
<td>Yes</td>
</tr>
<tr>
<td>QUERY_ACCELERATION</td>
<td>CURRENT QUERY ACCEL</td>
<td>DSN6SPRM</td>
<td>DSN1PS2</td>
<td>Yes</td>
</tr>
<tr>
<td>QUERY_ACCEL_WAITFORDATA</td>
<td>WAIT FOR DATA</td>
<td>DSN6SPRM</td>
<td>DSN1PS2</td>
<td>Yes</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>QUESCE PERIOD</td>
<td>DSN6ARVP</td>
<td>DSN1PA</td>
<td>Yes</td>
</tr>
<tr>
<td>RANDOMATT</td>
<td>RANDOM ATTACH</td>
<td>DSN6GRP</td>
<td>DSN1PK</td>
<td>Yes</td>
</tr>
<tr>
<td>REALSTORAGE_MANAGEMENT</td>
<td>MANAGE REAL STORAGE</td>
<td>DSN6SPRM</td>
<td>DSN1PE1</td>
<td>Yes</td>
</tr>
<tr>
<td>REALSTORAGE_MAX</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>REC_FASTREPLICATION</td>
<td>FAST RESTORE</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>RECALL</td>
<td>RECALL DATABASE</td>
<td>DSN6SPRM</td>
<td>DSN1PO</td>
<td>RECALL</td>
</tr>
<tr>
<td>RECALLD</td>
<td>RECALL DELAY</td>
<td>DSN6SPRM</td>
<td>DSN1PO</td>
<td>Yes</td>
</tr>
<tr>
<td>REPHAGE</td>
<td>CURRENT REFRESH AGE</td>
<td>DSN6SPRM</td>
<td>DSN1P8</td>
<td>Yes</td>
</tr>
<tr>
<td>REMOTE_COPY_SW_ACCEL</td>
<td>REMOTE COPY SW ACCEL</td>
<td>DSN6LOGP</td>
<td>DSN1P1</td>
<td>Yes</td>
</tr>
<tr>
<td>RENAMETABLE</td>
<td>RENAME TABLE</td>
<td>DSN6SPRM</td>
<td>DSN1TP2</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG_DROP_PBG_PARTS</td>
<td>REORG DROP PBG PARTS</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG_LIST_PROCESSING</td>
<td>REORG LIST PROCESSING</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG_MAPPING_DATABASE</td>
<td>REORG MAPPING DB</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG_PART_SORT_NPSI</td>
<td>REORG PART SORT NPSI</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG_TS_NOPAD_DEFAULT</td>
<td>REORG TS NOPAD DEFAULT</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>RESTART</td>
<td>RESTART OR DEFER</td>
<td>DSN6SPRM</td>
<td>DSN1TPS</td>
<td>RESTART</td>
</tr>
<tr>
<td>RESTORE_RECOVER_FROMDUMP</td>
<td>RESTORE/RECOVER</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>RESTORE_TAPEUNITS</td>
<td>MAXIMUM TAPE UNITS</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG_TS_NOPAD_DEFAULT</td>
<td>REORG TS NOPAD DEFAULT</td>
<td>DSN6SPRM</td>
<td>DSN1P6</td>
<td>Yes</td>
</tr>
<tr>
<td>RESTRICT_ALT_COL_FOR_DCC</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>RESYNC</td>
<td>RESYNC INTERVAL</td>
<td>DSN6FAC</td>
<td>DSN1PR</td>
<td>Yes</td>
</tr>
<tr>
<td>RETLWAIT</td>
<td>RETAINED LOCK TIMEOUT</td>
<td>DSN6SPRM</td>
<td>DSN1P1</td>
<td>Yes</td>
</tr>
<tr>
<td>RETRY_STOPPED_OBJECT</td>
<td>RETRY STOPPED OBJECTS</td>
<td>DSN6SPRM</td>
<td>DSN1P72</td>
<td>RSO</td>
</tr>
<tr>
<td>REVOKE_DEP_PRIVILEGES</td>
<td>REVOKE DEP PRIV</td>
<td>DSN6SPRM</td>
<td>DSN1P1</td>
<td>Yes</td>
</tr>
<tr>
<td>RGCFCOLID</td>
<td>REGISTRATION OWNER</td>
<td>DSN6SPRM</td>
<td>DSN1PZ</td>
<td>RGCFCOLID</td>
</tr>
<tr>
<td>RGFDBNAM</td>
<td>REGISTRATION DATABASE</td>
<td>DSN6SPRM</td>
<td>DSN1PZ</td>
<td>RGFDBNAM</td>
</tr>
<tr>
<td>RGFDEDPL</td>
<td>CONTROL ALL APPLICATIONS</td>
<td>DSN6SPRM</td>
<td>DSN1PZ</td>
<td>RGFDEDPL</td>
</tr>
</tbody>
</table>
Table 42. Directory of subsystem parameters and application default values (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTIDxx entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGFDEFLT</td>
<td>UNREGISTERED DDL DEFAULT</td>
<td>DSN6SPRM</td>
<td>DSNTIPZ</td>
<td></td>
<td>RGFDEFLT</td>
</tr>
<tr>
<td>RGFESCP</td>
<td>ART/ORT ESCAPE CHARACTER</td>
<td>DSN6SPRM</td>
<td>DSNTIPZ</td>
<td></td>
<td>RGFESCP</td>
</tr>
<tr>
<td>RGFFULLQ</td>
<td>REQUIRE FULL NAMES</td>
<td>DSN6SPRM</td>
<td>DSNTIPZ</td>
<td></td>
<td>RGFFULLQ</td>
</tr>
<tr>
<td>RGFINSTL</td>
<td>INSTALL DD CONTROL SUPT</td>
<td>DSN6SPRM</td>
<td>DSNTIPZ</td>
<td></td>
<td>RGFINSTL</td>
</tr>
<tr>
<td>RGFNMORT</td>
<td>OBJ REGISTRATION TABLE</td>
<td>DSN6SPRM</td>
<td>DSNTIPZ</td>
<td></td>
<td>RGFNMORT</td>
</tr>
<tr>
<td>RGFMNPRT</td>
<td>APPL REGISTRATION TABLE</td>
<td>DSN6SPRM</td>
<td>DSNTIPZ</td>
<td></td>
<td>RGFMNPRT</td>
</tr>
<tr>
<td>RLF</td>
<td>RLF AUTO START</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td></td>
<td>RLFSTART</td>
</tr>
<tr>
<td>RLFENABLE</td>
<td>RLF SCOPE</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td></td>
<td>RLFENABLE</td>
</tr>
<tr>
<td>RLFAUTH</td>
<td>RESOURCE AUTHID</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td>Yes</td>
<td>PROTRLF</td>
</tr>
<tr>
<td>RLFFERR</td>
<td>DYNAMIC SQL</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td>Yes</td>
<td>RLFFERR</td>
</tr>
<tr>
<td>RLFFERRD</td>
<td>REMOTE DYNAMIC SQL</td>
<td>DSN6FAC</td>
<td>DSNTIPO4</td>
<td>Yes</td>
<td>RLFFERRD</td>
</tr>
<tr>
<td>RLFFERRSTC</td>
<td>STATIC SQL</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td>Yes</td>
<td>RLFFERRST</td>
</tr>
<tr>
<td>RLFFERRDSTC</td>
<td>REMOTE STATIC SQL</td>
<td>DSN6FAC</td>
<td>DSNTIPO4</td>
<td>Yes</td>
<td>RLFFERRD</td>
</tr>
<tr>
<td>RLFTBL</td>
<td>RLST NAME SUFFIX</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td></td>
<td>RLFSUFF</td>
</tr>
<tr>
<td>ROUTCDE</td>
<td>WTO ROUTE CODES</td>
<td>DSN6SYSP</td>
<td>DSNTIPO4</td>
<td></td>
<td>OPROUTCDE</td>
</tr>
<tr>
<td>RRULOCK</td>
<td>U LOCK FOR RR/RS</td>
<td>DSN6SPRM</td>
<td>DSNTIP1</td>
<td>Yes</td>
<td>RRULOCK</td>
</tr>
<tr>
<td>SCCSID</td>
<td>EBCDIC CCSID</td>
<td>DSNHDECP</td>
<td>DSNTIPO1</td>
<td></td>
<td>SCSSID</td>
</tr>
<tr>
<td>SECADM1</td>
<td>SECURITY ADMIN 1</td>
<td>DSN6SPRM</td>
<td>DSNTIPO4</td>
<td>Yes</td>
<td>SECADM1</td>
</tr>
<tr>
<td>SECADM1_INPUT_STYLE</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>SECADM1_INPUT_STYLE</td>
</tr>
<tr>
<td>SECADM1_Type</td>
<td>SEC ADMIN 1 TYPE</td>
<td>DSN6SPRM</td>
<td>DSNTIPO1</td>
<td>Yes</td>
<td>SECADM1_Type</td>
</tr>
<tr>
<td>SECADM2</td>
<td>SECURITY ADMIN 2</td>
<td>DSN6SPRM</td>
<td>DSNTIPO1</td>
<td>Yes</td>
<td>SECADM2</td>
</tr>
<tr>
<td>SECADM2_INPUT_STYLE</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>SECADM2_INPUT_STYLE</td>
</tr>
<tr>
<td>SECADM2_Type</td>
<td>SEC ADMIN 2 TYPE</td>
<td>DSN6SPRM</td>
<td>DSNTIPO1</td>
<td>Yes</td>
<td>SECADM2_Type</td>
</tr>
<tr>
<td>SECQTY</td>
<td>SECONDARY QTY</td>
<td>DSN6ARVP</td>
<td>DSNTHPA</td>
<td>Yes</td>
<td>SECQTY</td>
</tr>
<tr>
<td>SEPARATE_SECURITY</td>
<td>SEPARATE SECURITY</td>
<td>DSN6SPRM</td>
<td>DSNTIPO1</td>
<td>Yes</td>
<td>SEPARATE_SECURITY</td>
</tr>
<tr>
<td>SIGNON_MODULE</td>
<td>SIGNON</td>
<td>DSN6SYSP</td>
<td>DSNTIPO3</td>
<td></td>
<td>SIGNON_MODULE</td>
</tr>
<tr>
<td>SIMULATED_CPU_COUNT</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>SIMCCNT</td>
</tr>
<tr>
<td>SIMULATED_CPU_SPEED</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>SIMCSPD</td>
</tr>
<tr>
<td>SITETYPE</td>
<td>SITE TYPE</td>
<td>DSN6SPRM</td>
<td>DSNTIPO</td>
<td></td>
<td>SITETYPE</td>
</tr>
<tr>
<td>SJTABLES</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>—</td>
<td>Yes</td>
<td>SJTABLES</td>
</tr>
<tr>
<td>SKIPUNCI</td>
<td>SKIP UNCOMM INSERTS</td>
<td>DSN6SPRM</td>
<td>DSNTIP8</td>
<td>Yes</td>
<td>SKIPUNCI</td>
</tr>
<tr>
<td>SMF89</td>
<td>—</td>
<td>DSN6SYSP</td>
<td>—</td>
<td>Yes</td>
<td>SMF89</td>
</tr>
<tr>
<td>SMFACCT</td>
<td>SMF ACCOUNTING</td>
<td>DSN6SYSP</td>
<td>DSNTIPN</td>
<td></td>
<td>SMFACCT</td>
</tr>
<tr>
<td>SMFCOMP</td>
<td>COMPRESS SMF RECS</td>
<td>DSN6SYSP</td>
<td>DSNTIPN</td>
<td>Yes</td>
<td>SMFCOMP</td>
</tr>
<tr>
<td>SMFSTAT</td>
<td>SMF STATISTICS</td>
<td>DSN6SYSP</td>
<td>DSNTIPN</td>
<td></td>
<td>SMFSTAT</td>
</tr>
<tr>
<td>SPT01_INLINE_LENGTH</td>
<td>SPT01 INLINE LENGTH</td>
<td>DSN6SPRM</td>
<td>DSNTIPA2</td>
<td>Yes</td>
<td>SPT01_INLINE_LENGTH</td>
</tr>
<tr>
<td>SQLDEL</td>
<td>SQL STRING DELIMITER</td>
<td>DSNHDECP</td>
<td>DSNTIPO</td>
<td></td>
<td>SQLDEL</td>
</tr>
<tr>
<td>SQLLEVEL</td>
<td>PRECOMPLIER SQL LEVEL</td>
<td>DSNHDECP</td>
<td>DSNTIPO0</td>
<td>Yes</td>
<td>SQLLEVEL</td>
</tr>
<tr>
<td>SRTPOOL</td>
<td>SORT POOL SIZE</td>
<td>DSN6SPRM</td>
<td>DSNTIPC</td>
<td>Yes</td>
<td>SRTPOOL</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Field name</td>
<td>Macro</td>
<td>Panel</td>
<td>Update online</td>
<td>DSNTID.LIST entry or z/OSMF variable</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------</td>
<td>---------------</td>
<td>-------</td>
<td>---------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>SID</td>
<td>SUBSYSTEM NAME</td>
<td>DSNHDECP</td>
<td>DSNTPM</td>
<td>Yes</td>
<td>MVSSNAME</td>
</tr>
<tr>
<td>STARJOIN</td>
<td>STAR JOIN QUERIES</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>STARJOIN</td>
</tr>
<tr>
<td>STATCLGSRT</td>
<td>STATISTICS COLGROUP</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>STCS</td>
</tr>
<tr>
<td>STATFDBK_PROFILE</td>
<td>STATISTICS PROFILE</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>SFPR</td>
</tr>
<tr>
<td>STATFDBK_SCOPE</td>
<td>STATISTICS FEEDBACK</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>STATHIST</td>
</tr>
<tr>
<td>STATTIME</td>
<td>STATISTICS TIME</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>OPSTATIM</td>
</tr>
<tr>
<td>STATPGSAMP</td>
<td>PAGE-LEVEL SAMPLING</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>No</td>
<td>STATPGSAMP</td>
</tr>
<tr>
<td>STATROLL</td>
<td>STATISTICS ROLLUP</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>STATROLL</td>
</tr>
<tr>
<td>STATSINT</td>
<td>REAL TIME STATS</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>STATSSINT</td>
</tr>
<tr>
<td>STDSQL</td>
<td>STD SQL LANGUAGE</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEFSQLVL</td>
</tr>
<tr>
<td>STORMXAB</td>
<td>MAX ABEND COUNT</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>STORMXAB</td>
</tr>
<tr>
<td>STORTIME</td>
<td>TIMEOUT VALUE</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>STORTIME</td>
</tr>
<tr>
<td>SUBQ_MIDX</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>--</td>
<td>Yes</td>
<td>SUBQ_MIDX</td>
</tr>
<tr>
<td>SUPERRS</td>
<td>SUPPRESS SOFT ERRORS</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>SUPERRS</td>
</tr>
<tr>
<td>SUPPRESS_HINT_SQLCODE_DYN</td>
<td>—</td>
<td>DSN6SPRM</td>
<td>--</td>
<td>Yes</td>
<td>SHSD</td>
</tr>
<tr>
<td>SYVOLARC</td>
<td>SINGLE VOLUME</td>
<td>DSN6ARVP</td>
<td>DSNTPA</td>
<td>Yes</td>
<td>SYVOLARC</td>
</tr>
<tr>
<td>SYNCSVAL</td>
<td>STATISTICS SYNC</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>SYNCSVAL</td>
</tr>
<tr>
<td>SYSADM</td>
<td>SYSTEM ADMIN 1</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>PROTADMN</td>
</tr>
<tr>
<td>SYSADM2</td>
<td>SYSTEM ADMIN 2</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>PROTADMN</td>
</tr>
<tr>
<td>SYSOPR1</td>
<td>SYSTEM OPERATOR 1</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>PROTOPR1</td>
</tr>
<tr>
<td>SYSOPR2</td>
<td>SYSTEM OPERATOR 2</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>PROTOPR2</td>
</tr>
<tr>
<td>SYSTEM_LEVEL_BACKUPS</td>
<td>SYSTEM-LEVEL BACKUPS</td>
<td>DSN6SPRM</td>
<td>DSNTP2</td>
<td>Yes</td>
<td>SYSLVLBK</td>
</tr>
<tr>
<td>TBSBP8K</td>
<td>DEFAULT 8-KB BUFFER POOL FOR USER DATA</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>TBSBP8K</td>
</tr>
<tr>
<td>TBSBP16K</td>
<td>DEFAULT 16-KB BUFFER POOL FOR USER DATA</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>TBSBP16K</td>
</tr>
<tr>
<td>TBSBP32K</td>
<td>DEFAULT 32-KB BUFFER POOL FOR USER DATA</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>TBSBP32K</td>
</tr>
<tr>
<td>TBSBPLOB</td>
<td>DEFAULT BUFFER POOL FOR USER LOB DATA</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>TBSBPLOB</td>
</tr>
<tr>
<td>TBSBPOOL</td>
<td>DEFAULT 4-KB BUFFER POOL FOR USER DATA</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>TBSBPOOL</td>
</tr>
<tr>
<td>TBSBXML</td>
<td>DEFAULT BUFFER POOL FOR USER XML DATA</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td>Yes</td>
<td>TBSBXML</td>
</tr>
<tr>
<td>TCPALVER</td>
<td>TCP/IP ALREADY VERIFIED</td>
<td>DSN6FAC</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>TCPALVER</td>
</tr>
<tr>
<td>TCPKPAVL</td>
<td>TCP/IP KEEPALIVE</td>
<td>DSN6FAC</td>
<td>DSNTP5</td>
<td>Yes</td>
<td>TCPKPAVL</td>
</tr>
<tr>
<td>TEMPLATE_TIME</td>
<td>TEMPLATE TIME</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>TPTM</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME FORMAT</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEFTIME</td>
</tr>
<tr>
<td>TIMELEN</td>
<td>LOCAL TIME LENGTH</td>
<td>DSNHDECP</td>
<td>DSNTP4</td>
<td>Yes</td>
<td>DEFTIME</td>
</tr>
<tr>
<td>TRACSTR</td>
<td>TRACE AUTO START</td>
<td>DSN6SPRM</td>
<td>DSNTPN</td>
<td>Yes</td>
<td>Optrcaut</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Field name</td>
<td>Macro</td>
<td>Panel</td>
<td>Update online</td>
<td>DSNTDXYZ entry or z/OSMF variable</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------</td>
<td>------------</td>
<td>---------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>TRACTBL</td>
<td>TRACE SIZE</td>
<td>DSN6SPRM</td>
<td>DSNTPN</td>
<td></td>
<td>CALC/OPTRCSIZ</td>
</tr>
<tr>
<td>TRKRSITE</td>
<td>TRACKER SITE</td>
<td>DSN6SPRM</td>
<td>DSNTP0</td>
<td></td>
<td>TRKRSITE</td>
</tr>
<tr>
<td>TS_COMPRESSION_TYPE</td>
<td>TS_COMPRESSION_TYPE</td>
<td>DSN6SPRM</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>--</td>
</tr>
<tr>
<td>TSQTY</td>
<td>TABLE SPACE ALLOCATION</td>
<td>DSN6SPY</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>TSQTY</td>
</tr>
<tr>
<td>TS stamp</td>
<td>TIMESTAMP ARCHIVE</td>
<td>DSN6ARVP</td>
<td>DSNTPH</td>
<td></td>
<td>ARCHTS</td>
</tr>
<tr>
<td>TWOACTV</td>
<td>NUMBER OF COPIES</td>
<td>DSN6LOGP</td>
<td>DSNTPH</td>
<td></td>
<td>CALC/ARCHTWO</td>
</tr>
<tr>
<td>TWOARCH</td>
<td>NUMBER OF COPIES</td>
<td>DSN6LOGP</td>
<td>DSNTPH</td>
<td></td>
<td>CALC/ARCHTWO</td>
</tr>
<tr>
<td>TWOBSDS</td>
<td></td>
<td>DSN6LOGP</td>
<td></td>
<td></td>
<td>TWOBSDS</td>
</tr>
<tr>
<td>UGCCSIDS</td>
<td>UNICODE CCSID</td>
<td>DSNHDECP</td>
<td>DSNTPF</td>
<td></td>
<td>UGCCSIDS</td>
</tr>
<tr>
<td>UIFCIDS</td>
<td>UNICODE IFIDS</td>
<td>DSN6SPRM</td>
<td>DSNTPN</td>
<td>Yes</td>
<td>UIFCIDS</td>
</tr>
<tr>
<td>UMCCSID</td>
<td>UNICODE CCSID</td>
<td>DSNHDECP</td>
<td>DSNTPF</td>
<td></td>
<td>UMCCSIDS</td>
</tr>
<tr>
<td>UNIO_RC1_COLNAME_2</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td></td>
<td>UNCOLN7</td>
</tr>
<tr>
<td>UNIT</td>
<td>DEVICE TYPE 1</td>
<td>DSN6ARVP</td>
<td>DSNTPA</td>
<td></td>
<td>ARCHDEVT</td>
</tr>
<tr>
<td>UNIT2</td>
<td>DEVICE TYPE 2</td>
<td>DSN6ARVP</td>
<td>DSNTPA</td>
<td></td>
<td>ARCHDEV2</td>
</tr>
<tr>
<td>URCHECKTH</td>
<td>UR CHECK FREQ</td>
<td>DSN6SPY</td>
<td>DSNTP1L</td>
<td>Yes</td>
<td>URCHECKTH</td>
</tr>
<tr>
<td>URLGWT</td>
<td>UR LOG WRITE CHECK</td>
<td>DSN6SPY</td>
<td>DSNTP1L</td>
<td></td>
<td>URLGWT</td>
</tr>
<tr>
<td>USCCSID</td>
<td>UNICODE CCSID</td>
<td>DSNHDECP</td>
<td>DSNTPF</td>
<td></td>
<td>USCCSID</td>
</tr>
<tr>
<td>DB2 BACKUP STG GROUP</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP62</td>
<td>Yes</td>
<td>UDBSG</td>
</tr>
<tr>
<td>DB LOG BACKUP STG GRP</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP62</td>
<td>Yes</td>
<td>ULSBG</td>
</tr>
<tr>
<td>HSM MESSAGE DS HLQ</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP62</td>
<td>Yes</td>
<td>UHMDH</td>
</tr>
<tr>
<td>DUMP TEMP STORCLAS</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td>Yes</td>
<td>UTTEMPSC</td>
</tr>
<tr>
<td>DUMP CLASS NAME</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP62</td>
<td>Yes</td>
<td>UTDUMPC1</td>
</tr>
<tr>
<td>UTILITY OBJECT_conversion</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP7</td>
<td>Yes</td>
<td>UTOC</td>
</tr>
<tr>
<td>UTIMOUT</td>
<td>UTILITY TIMEOUT</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td></td>
<td>UTIMOUT</td>
</tr>
<tr>
<td>UD膂DDEV</td>
<td>TEMP DS UNIT NAME</td>
<td>DSN6SPRM</td>
<td>DSNTP6</td>
<td></td>
<td>VOLTEDV</td>
</tr>
<tr>
<td>WFDBSEP</td>
<td>SEPARATE WORK FILES</td>
<td>DSN6SPRM</td>
<td>DSNTP91</td>
<td></td>
<td>WFDBSEP</td>
</tr>
<tr>
<td>WFSST</td>
<td>AGENT LEVEL THRESHOLD</td>
<td>DSN6SPRM</td>
<td>DSNTP91</td>
<td>Yes</td>
<td>WFSAT</td>
</tr>
<tr>
<td>DUMP TEMP THRESHOLD</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTP91</td>
<td>Yes</td>
<td>WFSST</td>
</tr>
<tr>
<td>WLM ENVIRONMENT</td>
<td></td>
<td>DSN6SPY</td>
<td>DSNTPX</td>
<td></td>
<td>WLMENV</td>
</tr>
<tr>
<td>XLKUPDLT</td>
<td>X LOCK FOR SEARCHED U/D</td>
<td>DSN6SPRM</td>
<td>DSNTP1</td>
<td></td>
<td>XLKUPDLT</td>
</tr>
<tr>
<td>XML_RANDOMIZE_DOCID</td>
<td>RANDOMIZE XML DOCID</td>
<td>DSN6SPY</td>
<td>DSNTP8</td>
<td></td>
<td>XRDI</td>
</tr>
<tr>
<td>XML_RESTRICT EMPTY_TAG</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td></td>
<td>NOET</td>
</tr>
<tr>
<td>ZOSMETRICS</td>
<td></td>
<td>DSN6SPRM</td>
<td></td>
<td></td>
<td>ZOSMETRIC</td>
</tr>
<tr>
<td>D82 zHyperLinks SCOPE</td>
<td></td>
<td>DSN6SPRM</td>
<td>DSNTPA3</td>
<td>Yes</td>
<td>ZHYPL</td>
</tr>
</tbody>
</table>
### Table 42. Directory of subsystem parameters and application default values (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTID.LCX entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>— — — — — —</td>
<td>ADVANCED CONFIGURATION OPTIONS</td>
<td>—</td>
<td>DSNTIPR1</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>APPL DEFAULTS</td>
<td>—</td>
<td>DSNTIP03</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>APPLICATION DBRM</td>
<td>—</td>
<td>DSNTIP1</td>
<td>DBRMLIB</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>APPLICATION LOAD</td>
<td>—</td>
<td>DSNTIP1</td>
<td>RUNLIB</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>ARCHIVE LOG FREQ</td>
<td>—</td>
<td>DSNTIP1</td>
<td>NUMHRARCC</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>AUTH MEMBER</td>
<td>—</td>
<td>DSNTIPM</td>
<td>MVSAMEMB</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>AUTH SEQUENCE</td>
<td>—</td>
<td>DSNTIPM</td>
<td>MVSASEEQN</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>BP0 - BP29</td>
<td>—</td>
<td>DSNTIP1</td>
<td>BP0 - BP29</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>BP30 - BP32K9</td>
<td>—</td>
<td>DSNTIP2</td>
<td>BP30 - BP32K9</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>BSDS AND ACTIVE LOG ICF CATALOG ALIAS</td>
<td>—</td>
<td>DSNTIPA4</td>
<td>BSDS and active log panel</td>
<td>VBSDSLOG</td>
</tr>
<tr>
<td>— — — — — —</td>
<td>BUFFER POOL SIZE</td>
<td>—</td>
<td>DSNTIPC</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>BYTE-BASED STRING FUNCTIONS</td>
<td>—</td>
<td>DSNTIPSA</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS LE PRELINKER PHASE</td>
<td>—</td>
<td>DSNTIPU</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS and IMS transaction invocation routines</td>
<td>—</td>
<td>DSNTIPRA</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS COBOL LIBRARY</td>
<td>—</td>
<td>DSNTIPW</td>
<td>CICSCOB</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS EXCI LIBRARY</td>
<td>—</td>
<td>DSNTIPW</td>
<td>CICSEXCI</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS LOAD LIBRARY</td>
<td>—</td>
<td>DSNTIPW</td>
<td>CICSMOD</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS MACRO LIBRARY</td>
<td>—</td>
<td>DSNTIPW</td>
<td>CICSMAC</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CICS PL/I LIBRARY</td>
<td>—</td>
<td>DSNTIPW</td>
<td>CICSPI1</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>C/CPP COMPILER LIBRARY</td>
<td>—</td>
<td>DSNTIPU</td>
<td>CCOMP</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>C/CPP COMPILER MODULE</td>
<td>—</td>
<td>DSNTIPU</td>
<td>—</td>
<td>CPROG</td>
</tr>
<tr>
<td>— — — — — —</td>
<td>C/CPP HEADER LIBRARY</td>
<td>—</td>
<td>DSNTIPU</td>
<td>—</td>
<td>CHDRS</td>
</tr>
<tr>
<td>— — — — — —</td>
<td>C/370 COMPILER MESSAGES</td>
<td>—</td>
<td>DSNTIPU</td>
<td>—</td>
<td>CMSGS</td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CLIST ALLOCATION</td>
<td>—</td>
<td>DSNTIPA2</td>
<td>VOLSDAT1</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>CLIST LIBRARY</td>
<td>—</td>
<td>DSNTIP1</td>
<td>OUTCL</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COBOL COMPILER LIBRARY</td>
<td>—</td>
<td>DSNTIPU1</td>
<td>ICOBCOMP</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COBOL LE PRELINKER PHASE</td>
<td>—</td>
<td>DSNTIPU1</td>
<td>ICOBPRLK</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COLUMNS</td>
<td>—</td>
<td>DSNTIPD</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COMMAND PREFIX</td>
<td>—</td>
<td>DSNTIPM</td>
<td>MVSSPREF</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COMMAND SCOPE</td>
<td>—</td>
<td>DSNTIPM</td>
<td>CMDSCOPES</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COPY 1 NAME</td>
<td>—</td>
<td>DSNTIPH</td>
<td>BSDSNAM1</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>COPY 2 NAME</td>
<td>—</td>
<td>DSNTIPH</td>
<td>BSDSNAM2</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>Active logs: COPY 1 PREFIX</td>
<td>—</td>
<td>DSNTIPH</td>
<td>LOGSREP1</td>
<td></td>
</tr>
<tr>
<td>— — — — — —</td>
<td>Active logs: COPY 2 PREFIX</td>
<td>—</td>
<td>DSNTIPH</td>
<td>LOGSREP2</td>
<td></td>
</tr>
<tr>
<td>Parameter name</td>
<td>Field name</td>
<td>Macro</td>
<td>Panel</td>
<td>Update online</td>
<td><strong>DSNTID</strong> entry or <strong>z/OSMF</strong> variable</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>-------</td>
<td>-------</td>
<td>---------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>CPPAUTCL</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>CPHDRS</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>CPPCLASS</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>CPPPROC</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>DSNLIBP</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>DSNLIBS</td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
<td>DSNTPU</td>
<td></td>
<td>AGNEWIDS</td>
</tr>
<tr>
<td>---</td>
<td>Databases</td>
<td></td>
<td>DSNTPD</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>Database metadata routines</td>
<td></td>
<td>DSNTPRA</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>Databases</td>
<td></td>
<td>DSNTEP</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>DBRM Library</td>
<td></td>
<td>DSNTPT</td>
<td></td>
<td>DSNDBRM</td>
</tr>
<tr>
<td>---</td>
<td>DB2 administrator routines</td>
<td></td>
<td>DSNTPRA</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>DB2 Base Java DIR</td>
<td></td>
<td>DSNTPX1</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>DB2 command prefix first character</td>
<td></td>
<td>DSNTPAP</td>
<td></td>
<td>AGSSIDPX</td>
</tr>
<tr>
<td>---</td>
<td>DB2 generic LUNAME</td>
<td></td>
<td>DSNTPR</td>
<td></td>
<td>GENLUNM</td>
</tr>
<tr>
<td>---</td>
<td>Target function level</td>
<td></td>
<td>DSNTP00</td>
<td></td>
<td>FUNCLVL</td>
</tr>
<tr>
<td>---</td>
<td>DB2 location name</td>
<td></td>
<td>DSNTPR</td>
<td></td>
<td>DDLOC</td>
</tr>
<tr>
<td>---</td>
<td>DB2 optimization routines</td>
<td></td>
<td>DSNTPRA</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>DB2 security routines</td>
<td></td>
<td>DSNTPRA</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>DB2 TCP/IP IPNAME</td>
<td></td>
<td>DSNTPR</td>
<td></td>
<td>IPNAME</td>
</tr>
<tr>
<td>---</td>
<td>DB2 VTAM LUNAME</td>
<td></td>
<td>DSNTPR</td>
<td></td>
<td>DDLUNAM</td>
</tr>
<tr>
<td>---</td>
<td>DB2 VTAM PASSWORD</td>
<td></td>
<td>DSNTPR</td>
<td></td>
<td>DDPASS</td>
</tr>
<tr>
<td>---</td>
<td>DD/MM/YYYY local date format</td>
<td></td>
<td>DSNTPA</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>---</td>
<td>Deadlock cycle</td>
<td></td>
<td>DSNTP1</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>Deadlock time</td>
<td></td>
<td>DSNTH1</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>Declaration library</td>
<td></td>
<td>DSNTP1</td>
<td></td>
<td>SRCLIB</td>
</tr>
<tr>
<td>---</td>
<td>Define catalog</td>
<td></td>
<td>DSNTH1</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>Disconnect IRLM</td>
<td></td>
<td>DSNTH1</td>
<td></td>
<td>IRLMDISC</td>
</tr>
<tr>
<td>---</td>
<td>English task and panels library</td>
<td></td>
<td>DSNTPAP</td>
<td></td>
<td>DSNPPFE</td>
</tr>
<tr>
<td>---</td>
<td>Enter job card information for install and sample jobs</td>
<td></td>
<td>DSNTP1</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td>Executed STMTS</td>
<td></td>
<td>DSNTPD</td>
<td></td>
<td>NUMSTMTE</td>
</tr>
<tr>
<td>---</td>
<td>Exit Library</td>
<td></td>
<td>DSNTH1</td>
<td></td>
<td>DSNEXIT</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Field name</td>
<td>Macro</td>
<td>Panel</td>
<td>Update online</td>
<td>DSNTID entry or z/OSMF variable</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>-------</td>
<td>-------</td>
<td>---------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td>FAVOR NEW ACCELERATOR TYPE</td>
<td></td>
<td>DSNTIP8A</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>FOR DB2 REXX ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVR</td>
</tr>
<tr>
<td></td>
<td>FOR DB2 UTILITIES ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVU</td>
</tr>
<tr>
<td></td>
<td>FOR DB2 XML PROCESSING ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVX</td>
</tr>
<tr>
<td></td>
<td>FOR MOST DB2 ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVG</td>
</tr>
<tr>
<td></td>
<td>FOR MOST DB2 JAVA ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVJ</td>
</tr>
<tr>
<td></td>
<td>FOR ROUTINES IDENTIFIED TO PROGRAM CONTROL</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVP</td>
</tr>
<tr>
<td></td>
<td>FOR SERIALIZED ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENV1</td>
</tr>
<tr>
<td></td>
<td>FOR THE DB2 CICS ROUTINE</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVC</td>
</tr>
<tr>
<td></td>
<td>FOR DB2 MQSERIES ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVM</td>
</tr>
<tr>
<td></td>
<td>FOR DB2 WEB SERVICES ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVW</td>
</tr>
<tr>
<td></td>
<td>FOR UNIFIED DEBUGGER ROUTINES</td>
<td></td>
<td>DSNTIPRI</td>
<td></td>
<td>WLMENVD</td>
</tr>
<tr>
<td></td>
<td>GDDM LOAD MODULES</td>
<td></td>
<td>DSNTIPW</td>
<td></td>
<td>GDDMLOAD</td>
</tr>
<tr>
<td></td>
<td>GDDM MACLIB</td>
<td></td>
<td>DSNTIPW</td>
<td></td>
<td>GDDMMACS</td>
</tr>
<tr>
<td></td>
<td>GRANT EXECUTE (on many sub-panels)</td>
<td></td>
<td>DSNTIPRA</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>GROUP ATTACH</td>
<td></td>
<td>DSNTIPK</td>
<td></td>
<td>GRPATTNM</td>
</tr>
<tr>
<td></td>
<td>HIGH LEVEL ASSEMBLER LIB</td>
<td></td>
<td>DSNTIPU</td>
<td></td>
<td>ASMLIB</td>
</tr>
<tr>
<td></td>
<td>IC DATA SET UNIT NAME</td>
<td></td>
<td>DSNTIPG1</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE DATA CLASS</td>
<td></td>
<td>DSNTIPG1</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE EXPRESSION</td>
<td></td>
<td>DSNTIPG1</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE MGMT CLASS</td>
<td></td>
<td>DSNTIPG1</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE STOR CLASS</td>
<td></td>
<td>DSNTIPG1</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IBM LE LINK EDIT LIB</td>
<td></td>
<td>DSNTIPU</td>
<td></td>
<td>LELKED</td>
</tr>
<tr>
<td></td>
<td>IBM LE PRELINK MSG LIB</td>
<td></td>
<td>DSNTIPU</td>
<td></td>
<td>LEPLMSGL</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY</td>
<td></td>
<td>DSNTIPU</td>
<td></td>
<td>LERUN</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY 2</td>
<td></td>
<td>DSNTIPU</td>
<td></td>
<td>LERUN2</td>
</tr>
<tr>
<td></td>
<td>IBM Text Search routines (DSNTIPR)</td>
<td></td>
<td>DSNTIPRA</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IMS RESLIB</td>
<td></td>
<td>DSNTIPW</td>
<td></td>
<td>RESLIB</td>
</tr>
<tr>
<td></td>
<td>INCLUDE LIBRARY</td>
<td></td>
<td>DSNTIP1</td>
<td></td>
<td>DSNCHDRS</td>
</tr>
</tbody>
</table>
### Table 42. Directory of subsystem parameters and application default values (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTIDxx entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INIT JAVA HEAP SIZE</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>INPUT MEMBER NAME</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>INSERT FROM SELECT</td>
<td>—</td>
<td>DSNTIPSA</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>INSTALL GRANTEE(S)</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>INSGRLST</td>
</tr>
<tr>
<td></td>
<td>INSTALL IC PREFIX</td>
<td>—</td>
<td>DSNTIPG1</td>
<td>—</td>
<td>INSICPRE</td>
</tr>
<tr>
<td></td>
<td>INSTALL IRLM</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>INSTALL DS PREFIX</td>
<td>—</td>
<td>DSNTIPG1</td>
<td>—</td>
<td>INSDSPRE</td>
</tr>
<tr>
<td></td>
<td>INSTALL PKG OWNER</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>INSPKOWN</td>
</tr>
<tr>
<td></td>
<td>INSTALL SQL ID</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>INSQLID</td>
</tr>
<tr>
<td></td>
<td>INSTALL TYPE</td>
<td>—</td>
<td>DSNTIPA1</td>
<td>—</td>
<td>INSTYPE</td>
</tr>
<tr>
<td></td>
<td>IRLM LOAD LIBRARY</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>DSNIRLM</td>
</tr>
<tr>
<td></td>
<td>IRLM XCF GROUP NAME</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>IRLMGRPAN</td>
</tr>
<tr>
<td></td>
<td>ISPF COMMAND TABLE LIBRARY</td>
<td>—</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>DSNISPFT</td>
</tr>
<tr>
<td></td>
<td>ISPF INSTALL TASK AND PANELS LIBRARY</td>
<td>—</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>DSNISPFP</td>
</tr>
<tr>
<td></td>
<td>ISPF ISPLINK MODULE</td>
<td>—</td>
<td>DSNTIPW</td>
<td>—</td>
<td>ISLOAD</td>
</tr>
<tr>
<td></td>
<td>ISPF MESSAGE LIBRARY</td>
<td>—</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>DSNISPFM</td>
</tr>
<tr>
<td></td>
<td>ISPF SKELETON LIBRARY</td>
<td>—</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>DSNISPS</td>
</tr>
<tr>
<td></td>
<td>IVP DATA LIBRARY</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>DSNIVPD</td>
</tr>
<tr>
<td></td>
<td>IVP GRANTEE</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>IVP PKG OWNER</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>IVPGRLST</td>
</tr>
<tr>
<td></td>
<td>IVP SQL ID</td>
<td>—</td>
<td>DSNTIPG</td>
<td>—</td>
<td>IVPPKOWN</td>
</tr>
<tr>
<td></td>
<td>Java and SQL procedure processing routines [DSNTIPRA]</td>
<td>—</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>IVPSQLID</td>
</tr>
<tr>
<td></td>
<td>Java ENV DATA SET</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Java ENV VAR FILE</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>JDBC AND SQL DLLS</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>DSNLOD2</td>
</tr>
<tr>
<td></td>
<td>JDBC/SQ LENV VAR DIR</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>JDBC/SQ L INSTALL ID</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>JSON routines [DSNTIPRA]</td>
<td>—</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>JVM ADDRESSING MODE</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>JVM PROPERTIES FILE</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LIBRARY NAME PREFIX</td>
<td>—</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>DSNTLIBP</td>
</tr>
<tr>
<td></td>
<td>LIBRARY NAME SUFFIX</td>
<td>—</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>DSNTLIBS</td>
</tr>
<tr>
<td></td>
<td>LICENSE TERMS ACCEPTED</td>
<td>—</td>
<td>DSNTIOPO2</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>LINK LIST ENTRY</td>
<td>—</td>
<td>DSNTIPM</td>
<td>—</td>
<td>MVSLSMEMB</td>
</tr>
<tr>
<td></td>
<td>LINK LIST LIBRARY</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>DSNLNK</td>
</tr>
<tr>
<td></td>
<td>LINK LIST SEQUENCE</td>
<td>—</td>
<td>DSNTIPM</td>
<td>—</td>
<td>MVSLSSEQN</td>
</tr>
<tr>
<td></td>
<td>LOAD LIBRARY</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>DSNLOAD</td>
</tr>
<tr>
<td></td>
<td>LOCK ENTRY SIZE</td>
<td>—</td>
<td>DSNTIPX1</td>
<td>—</td>
<td>IRLMLKSZ</td>
</tr>
</tbody>
</table>
Table 42. Directory of subsystem parameters and application default values  (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTID entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2</td>
<td>—</td>
<td>DSNTIPA4</td>
<td></td>
<td>VOLSDAT6 DATACLS6 MGMTCLS6</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1</td>
<td>—</td>
<td>DSNTIPA4</td>
<td></td>
<td>VOLSDAT7 DATACLS7 MGMTCLS7</td>
</tr>
<tr>
<td></td>
<td>MBCS EBCDIC</td>
<td>—</td>
<td>DSNTIP8A</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEMBER IDENTIFIER</td>
<td>—</td>
<td>DSNTIPJ</td>
<td></td>
<td>IRLMIDEN</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA</td>
<td>—</td>
<td>DSNTIPA3</td>
<td></td>
<td>VOLSDAT2 DATACLS2 MGMTCLS2</td>
</tr>
<tr>
<td></td>
<td>MACRO LIBRARY</td>
<td>—</td>
<td>DSNTIP1</td>
<td></td>
<td>DSNMACS</td>
</tr>
<tr>
<td></td>
<td>MAX APPL HEAP SIZE</td>
<td>—</td>
<td>DSNTIPX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAX STORAGE FOR LOCKS</td>
<td>—</td>
<td>DSNTIPJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MQseries routines (DSNTIPRO)</td>
<td>—</td>
<td>DSNTIPRA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NUMBER OF LOCK ENTRIES</td>
<td>—</td>
<td>DSNTIPJ</td>
<td></td>
<td>IRLMLTE</td>
</tr>
<tr>
<td></td>
<td>NUMBER OF LOGS</td>
<td>—</td>
<td>DSNTIPL</td>
<td></td>
<td>LOGSNUM</td>
</tr>
<tr>
<td></td>
<td>NUMBER OF TCBS</td>
<td>—</td>
<td>DSNTIPX</td>
<td></td>
<td>STORTCBS</td>
</tr>
<tr>
<td></td>
<td>OTC LICENSE USAGE</td>
<td>—</td>
<td>DSNTIPO1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OUTPUT MEMBER NAME</td>
<td>—</td>
<td>DSNTIPA1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PACKAGE LISTS</td>
<td>—</td>
<td>DSNTIPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GRANT OWNER (on many sub-panels)</td>
<td>—</td>
<td>DSNTIPRA(sub-panels)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PACKAGE STATEMENTS</td>
<td>—</td>
<td>DSNTIPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PACKAGES</td>
<td>—</td>
<td>DSNTIPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAGE PROTECT</td>
<td>—</td>
<td>DSNTIPJ</td>
<td></td>
<td>IRLMPGPR</td>
</tr>
<tr>
<td></td>
<td>PARAMETER MODULE</td>
<td>—</td>
<td>DSNTIPO3</td>
<td></td>
<td>DSNZPARAM</td>
</tr>
<tr>
<td></td>
<td>PERMANENT UNIT NAME</td>
<td>—</td>
<td>DSNTIPA3</td>
<td></td>
<td>VOCNDEVT</td>
</tr>
<tr>
<td></td>
<td>PLAN STATEMENTS</td>
<td>—</td>
<td>DSNTIPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLANS</td>
<td>—</td>
<td>DSNTIPD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PL/I COMPILER LIBRARY</td>
<td>—</td>
<td>DSNTIPO1</td>
<td></td>
<td>PLICOMP</td>
</tr>
<tr>
<td></td>
<td>PL/I LE PRELINKER PHASE</td>
<td>—</td>
<td>DSNTIPO1</td>
<td></td>
<td>PLIPRLK</td>
</tr>
<tr>
<td></td>
<td>PROVISION EXEC ID</td>
<td>—</td>
<td>DSNTIPD</td>
<td></td>
<td>AGEXECID</td>
</tr>
<tr>
<td></td>
<td>REMOTE LOCATION</td>
<td>—</td>
<td>DSNTIPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RESYNC PORT</td>
<td>—</td>
<td>DSNTIPR</td>
<td>RESYNPRT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ROUTINES CREATOR</td>
<td>—</td>
<td>DSNTIPG</td>
<td>AUTHID</td>
<td></td>
</tr>
</tbody>
</table>

410 Installation and Migration Guide
### Table 42. Directory of subsystem parameters and application default values (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>DSNTID entry or z/OSMF variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime statistics processing routines (DSNTIPRD)</td>
<td>SAMPLE LIBRARY</td>
<td>DSNTIP7</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>NEWSAMP</td>
</tr>
<tr>
<td>Secure port</td>
<td>SEC PORT</td>
<td>DSNTIP8</td>
<td>DSNTIPD</td>
<td>—</td>
<td>SECPORT</td>
</tr>
<tr>
<td>Secure def creator</td>
<td>SEC DEF CREATOR</td>
<td>DSNTIP7</td>
<td>DSNTIPG</td>
<td>—</td>
<td>SECDEFID</td>
</tr>
<tr>
<td>Start IRLM CTRACE</td>
<td>START IRLM CTRACE</td>
<td>DSNTIP7</td>
<td>DSNTIP7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Software service instance name prefix</td>
<td>SOFTWARE SERVICE INSTANCE NAME PREFIX</td>
<td>DSNTIP7</td>
<td>DSNTIPAP</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Sort library</td>
<td>SORT LIBRARY</td>
<td>DSNTIP7</td>
<td>DSNTIPW</td>
<td>—</td>
<td>SORTLIB</td>
</tr>
<tr>
<td>Subgrp attach</td>
<td>SUBGRP ATTACH</td>
<td>DSNTIP8</td>
<td>DSNTIPK</td>
<td>—</td>
<td>SCRPATNM</td>
</tr>
<tr>
<td>Subsystem member</td>
<td>SUBSYSTEM MEMBER</td>
<td>DSNTIP7</td>
<td>DSNTIPM</td>
<td>—</td>
<td>MVSSMEMB</td>
</tr>
<tr>
<td>Subsystem sequence</td>
<td>SUBSYSTEM SEQUENCE</td>
<td>DSNTIP7</td>
<td>DSNTIPM</td>
<td>—</td>
<td>MVSSSEQN</td>
</tr>
<tr>
<td>System maclib</td>
<td>SYSTEM MACLIB</td>
<td>DSNTIP7</td>
<td>DSNTIPW</td>
<td>—</td>
<td>MACLIB</td>
</tr>
<tr>
<td>System procedures</td>
<td>SYSTEM PROCEDURES</td>
<td>DSNTIP7</td>
<td>DSNTIPW</td>
<td>—</td>
<td>PROCLIB</td>
</tr>
<tr>
<td>System time temporal queries</td>
<td>SYSTEM_TIME TEMPORAL QUERIES</td>
<td>DSNTIP7</td>
<td>DSNTIPS8</td>
<td>—</td>
<td>DSNTIPW</td>
</tr>
<tr>
<td>Tables</td>
<td>TABLES</td>
<td>DSNTIP7</td>
<td>DSNTIPD</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Tables in stmt</td>
<td>TABLES IN STMT</td>
<td>DSNTIP7</td>
<td>DSNTIPD</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Tablespace</td>
<td>TABLESPACE</td>
<td>DSNTIP7</td>
<td>DSNTIPD</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Tcp/ip port</td>
<td>TCP/IP PORT</td>
<td>DSNTIP7</td>
<td>DSNTIPF</td>
<td>—</td>
<td>TCPORFIG</td>
</tr>
<tr>
<td>Temp clist library</td>
<td>TEMP CLIST LIBRARY</td>
<td>DSNTIP7</td>
<td>DSNTIP7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Temporary unit name</td>
<td>TEMPORARY UNIT NAME</td>
<td>DSNTIP7</td>
<td>DSNTIP3</td>
<td>—</td>
<td>TVOLDEVT</td>
</tr>
<tr>
<td>Trace formatting routines (DSNTIPRD)</td>
<td>Trace formatting routines</td>
<td>DSNTIP7</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Ts cols with precision of 12</td>
<td>TS COLS WITH PRECISION</td>
<td>DSNTIP7</td>
<td>DSNTIP8A</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Update rate</td>
<td>UPDATE RATE</td>
<td>DSNTIP7</td>
<td>DSNTIP1</td>
<td>—</td>
<td>NUMCOMHR</td>
</tr>
<tr>
<td>Unified debugger for Java and SQL routines (DSNTIPRD)</td>
<td>Unified debugger for Java and SQL routines</td>
<td>DSNTIP7</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Use ZOSMF workflow</td>
<td>USE ZOSMF WORKFLOW</td>
<td>DSNTIP7</td>
<td>DSNTIPA1</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Utility invocation routines (DSNTIPRD)</td>
<td>Utility invocation routines</td>
<td>DSNTIP7</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>VSAM catalog, default and work file database</td>
<td>VSAM CATALOG, DEFAULT AND WORK FILE DATABASE</td>
<td>DSNTIP7</td>
<td>DSNTIPA3</td>
<td>—</td>
<td>VOLSDAT3 DATACL5S3 MGMTCL5S3 STORCL5S3</td>
</tr>
<tr>
<td>Views</td>
<td>VIEWS</td>
<td>DSNTIP7</td>
<td>DSNTIPD</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Web services/soap routines (DSNTIPRD)</td>
<td>Web services/soap routines</td>
<td>DSNTIP7</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Wlm env name (on many sub-panels)</td>
<td>WLM ENV NAME (on many sub-panels)</td>
<td>DSNTIP7</td>
<td>DSNTIPRA</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Wlm proc name</td>
<td>WLM PROC NAME</td>
<td>DSNTIP7</td>
<td>DSNTIPX</td>
<td>—</td>
<td>STORWLM</td>
</tr>
</tbody>
</table>

Chapter 4. Tailoring Db2 12 installation and migration jobs with the CLIST 411
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Field name</th>
<th>Macro</th>
<th>Panel</th>
<th>Update online</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORK FILE DB</td>
<td>—</td>
<td>—</td>
<td>DSNTIPK</td>
<td>DSNTID entry or z/OSMF variable</td>
<td>1. For use only when migrating or installing Db2 by using z/OSMF workflows. For more information, see Automating Db2 installation or migration with z/OS Management Facility.</td>
</tr>
<tr>
<td>XML schema processing routines</td>
<td>—</td>
<td>DSNTIPRA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z/OS SDK HOME DIR</td>
<td>—</td>
<td>DSNTIPX1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z/OS USS Java CLASSES</td>
<td>—</td>
<td>DSNTIPX1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z/OS USS Java RUNTIME</td>
<td>—</td>
<td>DSNTIPX1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4K SORT PRIMARY SPACE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4K SORT WORK TS'S</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4K SORT WORK SEG SIZE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4K TEMP PRIMARY SPACE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4K TEMP WORK TS'S</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4K TEMP WORK SEG SIZE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32K SORT PRIMARY SPACE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32K SORT WORK TS'S</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32K SORT WORK SEG SIZE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32K TEMP PRIMARY SPACE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32K TEMP WORK TS'S</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32K TEMP WORK SEG SIZE</td>
<td>—</td>
<td>DSNTIP9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

1. For use only when migrating or installing Db2 by using z/OSMF workflows. For more information, see Automating Db2 installation or migration with z/OS Management Facility.
2. You must use a privilege set of the process that includes installation SYSADM authority or SECADM authority to update this parameter.
3. This subsystem parameter can only be changed online. Restarting Db2 has no effect on the setting.
4. For online updates the SQLLEVEL, the precompiler and coprocessor begin using the new value immediately. However, the SYSPRM.DEF_SQLLEVEL global variable value is not updated until Db2 is restarted.

**Related concepts:**
The scope and uniqueness of Db2 subsystem parameters in data sharing
Chapter 5. Installing Db2

Installing Db2 involves running the installation jobs, connecting the facilities that allow TSO, batch, IMS, and CICS to access Db2 resources, and preparing Db2 for use. After you complete the installation steps, function level 500 or higher is activated and available for use.

Before you begin

Before you begin installing Db2:

- Review the information about Db2 security with your system security administrator, and plan and implement your Db2 security environment. See in particular Defining RACF user IDs for Db2-started tasks (Managing Security) and Permitting RACF access (Managing Security) which can help you plan your security environment even if you use an alternative security product. After initial installation of Db2, subsequent installation tasks and jobs require specific authorization in order to connect to Db2.
- Complete the SMP/E steps, load the Db2 libraries, and run the installation CLIST to tailor the Db2 jobs to your environment.
- Define an SMS environment for Db2. At a minimum, the data sets for the Db2 catalog and directory must be SMS-managed. For information on using sample job DSNTIJSS to define an SMS environment for data sets that are associated with the Db2 catalog and directory, see Installation step 1: Define the SMS environment for Db2 catalog and directory data sets: DSNTIJSS.
- If you plan to have data sharing enabled (Enable option on panel DSNTIPP1), you should save all your jobs from the original installation or migration in a different data set than the enabling jobs. If you save them in prefix.NEW.SDSNSAMP, the jobs might be overwritten.
- See the Db2 Program Directory for keyword specifications used for Preventive Service Planning (PSP). Use Information/Access or the ServiceLink facility of IBMLink to check the most current information about Db2 and other products. Contact IBM Support if you do not have access to IBMLink.

You must not use secondary authorization IDs to perform any of the following installation steps.

Procedure

After you complete the SMP/E steps and run the installation CLIST, you can edit and run the jobs that install your Db2 Db2 12 subsystem. Complete the following tasks:

“Installation step 1: Define the SMS environment for Db2 catalog and directory data sets: DSNTIJSS” on page 414
“Installation step 2: Define Db2 to IRLM and MVS: DSNTIJMV” on page 415
“Installation step 3: Define Db2 address space startup procedures, language procedures, and the Java WLM environment: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ” on page 419
“Installation step 4: Define the ICF catalog and alias: DSNTIJCA (optional)” on page 421
“Installation step 5: Define system data sets: DSNTIJIN” on page 422
“Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL” on page 424
“Installation step 7: Initialize system data sets: DSNTIJID, DSNTIJIE, and DSNTIJIF” on page 425
“Installation step 8: Define user authorization exit routines: DSNTIJEX (optional)” on page 426
“Installation step 9: Record Db2 data to SMF (optional)” on page 427
“Installation step 10: Establish subsystem security (optional)” on page 429
“Installation step 11: Connect Db2 to TSO” on page 430
“Installation step 12: Connect Db2 to IMS (optional)” on page 437
“Installation step 13: Connect CICS to Db2 (optional)” on page 437
“Installation step 14: Define Db2 to z/OS” on page 437
“Installation step 15: Start the Db2 subsystem” on page 438
“Installation step 16: Tailor the Db2 catalog: DSNTIJTC” on page 440
“Installation step 17: Activate Db2 12 function levels: DSNTIJAF” on page 441
“Installation step 18: Create default storage group, define temporary work files, and bind Db2 REXX Language Support: DSNTIJTM” on page 441
“Installation step 19: Define and bind Db2 objects: DSNTIJSO” on page 443
“Installation step 20: Populate the user-maintained databases (optional)” on page 445
“Installation step 21: Configure Db2 for running stored procedures and user-defined functions” on page 445
“Installation step 22: Set up Db2-supplied routines” on page 457
“Installation step 23: Back up the Db2 directory and catalog: DSNTIJIC” on page 469
“Installation step 24: Set up the administrative task scheduler” on page 470
“Installation step 25: Verify successful installation of Db2 12” on page 473

What to do next

You might want to enable more capabilities in the newly installed Db2 subsystem.
For instructions, see Working with additional capabilities for Db2

Related concepts:
Loading the Db2 libraries

Related tasks:
Strategy for creating a Db2 data sharing group
Tailoring Db2 12 installation and migration jobs with the CLIST
Automating Db2 migration by using z/OS Management Facility

Related information:

Installation step 1: Define the SMS environment for Db2 catalog and directory data sets: DSNTIJSS

All data sets for the Db2 catalog and directory must reside on SMS-managed storage and belong to a SMS data class that is defined with the extended addressability attribute.
About this task

To define a data class with the extended addressability attribute, specify EXT in the DATA SET NAME TYPE field of the DATA SET CLASS DEFINE panel of ISMF and ensure that the automatic class selection (ACS) routine associates the Db2 catalog and directory data sets with this data class. Job DSNTIJSS shows how to create a stand-alone SMS environment for the Db2 catalog and directory data sets. DSNTIJSS is designed for use on systems that do not already have a SMS environment, but it can also be used as a reference for adapting an existing one. In particular, job step STEP061 shows an example of defining a data class with the extended addressability attribute, and job step STEP071 shows example ACS routines that associate the Db2 catalog and directory data sets with the data class and a storage class.

Procedure

To create the stand-alone SMS environment for the Db2 catalog and directory:
1. Customize job DSNTIJSS according to the directions in the job prolog.
2. Run job DSNTIJSS.
3. To activate the SMS environment, use this z/OS command, where scds-name is the name of the SMS source control data set that was specified by DSNTIJSS:
   SETSMS SCDS( scds-name )

   Attention: This command will deactivate any existing SMS environment that is defined from another SCDS.

Installation step 2: Define Db2 to IRLM and MVS: DSNTIJMV

You must update certain z/OS SYS1.PARMLIB members to identify Db2 to IRLM and MVS.

Before you begin

The z/OS operating system must be installed at the prerequisite level.

About this task

Job DSNTIJMV demonstrates how to update the SYS1.PARMLIB members to identify Db2 to IRLM and MVS. Although you might edit and run this job, the more common approach is to examine the changes that DSNTIJMV makes and use an editor, such as ISPF/PDF, to make the required updates directly to the SYS1.PARMLIB members.

Because DSNTIJMV cannot anticipate all the necessary updates to identify Db2 to MVS and IRLM in your z/OS environment, additional updates might be required for your site. You can complete these updates either by making the updates directly in SYS1.PARMLIB members, or by editing the DSNTIJMV job.

Do not overwrite the entries for Db2 and IRLM in the program properties table (PPT). Note that PPT entries should no longer be defined in SYS1.PARMLIB(SCHEDxx). Default entries are supplied in the SYS1.LINKLIB(IEFSRDPT) load module, which is provided by z/OS. For details of the PPT entries, see Program properties table (PPT) (MVS Initialization and Tuning Reference).
Procedure

To define Db2 to IRLM and MVS:

1. Examine the DSNTIJMV job and study the updates that it makes.
2. Update the following SYS1.PARMLIB members by editing them directly, or by tailoring and running job DSNTIJMV:

**IEFSSNxx**
Contains an entry for every z/OS subsystem. Db2 adds one entry for Db2 and two entries for IRLM in the list. The second IRLM entry, whose subsystem name is JRLM, simplifies adding a second IRLM to your system if the first is damaged. You must provide your own procedure to add JRLM. Unique names must be used for each entry.

z/OS provides subsystem entries for Db2 and IRLM in IEFSSN00. Examine these entries to determine whether they are appropriate for your needs. Make sure that a subsystem name appears only once in the subsystem name list.

Place the SMS line as the first line in an IEFSSNxx member. Place the primary system’s record (JES2 or JES3) record next. Other products might affect the position of the entries during system initialization. The Db2 line should come after SMS, the JES subsystem, and other vendor products.

The Db2 entry has the following format:

```plaintext
SUBSYS SUBNAME(ssname) INITRTN(DSN3INI)
INITPARM('DSN3EPX,cmd-prefix<,scope<,group-attach<<,subgroup-attach>>>)'
```

- **ssname**
  The Db2 subsystem name.

- **DSN3INI**
  The name of the Db2 load module that z/OS invokes during master scheduler initialization. This module must be located in a link list data set (or in SYS1.LINKLIB).

- **DSN3EPX**
  The name of the Db2 load module that responds to Db2 requests that are received from the z/OS subsystem interface. (Db2 can be active or inactive when the requests are received.) This module must be located in a link list data set (or in SYS1.LINKLIB).

- **cmd-prefix**
  The one- to eight-character command prefix.

  The first character of the command prefix must be one of the following characters: @ # . / ` * + - = $ < ! & % _ ? : " . The remaining characters of the command prefix must be one of the above characters, A-Z, or 0-9. Do not use the JES2 backspace character or command prefix character. The default is the hyphen (-).

  Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or z/OS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix beginning from the first character. For example, you cannot assign ' - ' to one subsystem and ' -Db2A ' to
another. Similarly, you also cannot assign `?DB2` to one subsystem and `?DB2A` to another. You can assign `-DB2A` and `-DB2B` to different Db2 subsystems.

**scope**

The one-character scope for the command prefix. Db2 registers its command prefix with the operating system. When this is done, the scope of the command prefix is controlled by the value you choose:

- **S** Started; and the prefix is registered with Sysplex scope at Db2 startup instead of during z/OS IPL. This is the default.

  *Recommendation:* Choose S, which allows you to have a single IEFSSNxx PARMLIB member that all z/OS systems in the Sysplex can use. Choosing S also simplifies the task of moving a Db2 from one system to another; you can stop Db2 on one z/OS and start it up on another without having to re-IPL the system.

- **M** z/OS system scope; the prefix is registered during z/OS IPL.

- **X** Sysplex scope; the prefix is registered during z/OS IPL. As a result, this Db2 cannot be restarted on another z/OS without changing the definitions and re-IPLing both z/OS systems.

**group-attach**

The group attachment name, which is used for data sharing. You can specify this on installation panel DSNTIPK.

**subgroup-attach**

is the name of the subgroup attachment, which is used in data sharing environments to provide more control over how TSO, CAF, RRSAF, JDBC, ODBC, and Db2 utilities connections find active Db2 subsystems. You can specify this value by entering a subgroup name in the SUBGRP ATTACH field on installation panel DSNTIPK.

**IEAAPFxx or PROGxx**

Job DSNTIJMV updates IEEAPExx to include the Db2 program libraries (`prefix.SDSNEXIT, prefix.SDSNLOAD, prefix.SDSNL0D2, prefix.SDXRRESL, and prefix.SDSNLINK`) as APF-authorized libraries.

If a program library that contains DFSORT or Db2 Sort for z/OS is not already APF-authorized, you can edit DSNTIJMV to authorize it. To do this, you can include the authorization either in this list or in LNKSTxx. All libraries that are concatenated with `prefix.SDSNLOAD` in STEPLIB and JOBLIB statements must be APF-authorized. Ensure that the volume serial number in this member is the volume on which the data set resides.

If you are using the PROGxx member instead of the IEEAPExx member, you must update the member manually—job DSNTIJMV does not edit it.

**LNKLSTxx**

Job DSNTIJMV updates this member to include the Db2 load module library, `prefix.SDSNLINK`. If you moved the modules from `prefix.SDSNLINK` into another library, edit DSNTIJMV to include that
library in the LNKLSTxx member. If you have combined prefix.SDSNLINK and prefix.SDSNLOAD into one library, edit DSNTIJMV to include the combined library in the LNKLSTxx member.

Any data set that is added to the LNKLSTxx member must be cataloged in the master catalog of the system. This is normally true of prefix.SDSNLINK; however, if an alias points to a user catalog when you run DSNALLOC, prefix.SDSNLINK is cataloged in a user catalog. In this case, you must either ensure that prefix.SDSNLINK is also cataloged in the master catalog or give prefix.SDSNLINK a high-level qualifier other than prefix, the high-level qualifier for this release. You must give a high-level qualifier other than prefix to all release-sensitive data sets that are placed in the LNKLSTxx member.

If you do not include the DFSORT library, the Db2 Sort for z/OS library, or both in the LNKLSTxx member, you must provide a JOBLIB or STEPLIB statement for all utility jobs that include one of those libraries. You can accomplish this by placing a STEPLIB statement in DSNUPROC, which appears later in this job. If you use customized modules and exits, prefix.SDSNEXIT must precede prefix.SDSNLOAD in JOBLIB and STEPLIB statements.

3. If necessary, update the NSYSLX option of the z/OS IEASYSxx PARMLIB member to define more z/OS system linkage indexes (LXs). Each Db2 and each IRLM that you define to z/OS in the IEFSSNxx PARMLIB member requires a z/OS system linkage index (LX). The default number of these indexes that z/OS reserves is 165. If you place all of your Db2 and IRLM subsystem definitions in a single IEFSSNxx member, you might need more than 165 LXs; otherwise, your subsystems might not start.

4. Complete the following steps to enable diagnosis of IRLM problems:
   a. Specify the IRLM dump formatting modules DXRRLM50, DXRRLM55, and DXRRLS55 in a BLSUSER member in a data set that is in the system’s PARMLIB concatenation.
   b. Copy the DXRRL186, DXRRRLFTB, DXRRLM50, DXRRLM55, and DXRRLS55 modules from the IRLM SDXRRESL data set to a data set in the system’s link list. Or, add the SDXRRESL data set to the system’s link list in a PROGxx member of a data set in the system’s PARMLIB concatenation. If you use more than one level of IRLM, you must use a STEPLIB concatenation when using IPCS to process dumps that are taken for the levels whose modules are not in the link list.

Related concepts:
- How Db2 assigns I/O priorities (Db2 Performance)

Related reference:
- COMMAND PREFIX field

Related information:
- Using the command prefix facility (z/OS MVS Planning: Operations)
Installation step 3: Define Db2 address space startup procedures, language procedures, and the Java WLM environment: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ

Before you begin

The authorization ID that runs the jobs must have authority to update the target system PROCLIB. You can enter that library name on installation panel DSNTIPW in the SYSTEM PROCEDURES field. You might want to use a procedure library other than SYS1.PROCLIB for the procedures.

The authorization ID that runs job DSNTIJMJ must have authority to create the data set and HFS files that the job specifies.

Examine the updates from the jobs in this step carefully before you run them. Also, you might have additional procedures of your own to rename.

About this task

Because different sites have different requirements, the jobs in the following steps cannot anticipate all the necessary updates to the target system procedure library in your z/OS environment. You might have additional procedures of your own to rename. You can complete these updates either by making the updates directly in the target system procedure library, or by editing the jobs in the following steps.

Procedure

To update the system procedure library for the target z/OS system for Db2, complete the following steps:

1. Add the Db2 and IRLM address space startup procedures by tailoring and running job DSNTIJMA. Job DSNTIJMA adds startup procedures in the target system procedure library for the following Db2 address spaces, where ssnm is the subsystem name:
   - ssnmMSTR for Db2 system services
   - ssnmDBM1 for Db2 database services
   - ssnmDIST for the Db2 distributed data facility (DDF)
   - ssnmADMT for the Db2 administrative task scheduler
   - IRLMPROC, or a user-defined procedure for IRLM

Examine the size of the private area on the Db2 startup procedures. If necessary, modify the procedures to satisfy the requirements for environmental descriptor manager (EDM) pool size, buffers, number of data sets open, and amount of available private address space. For more information, see Calculating working storage requirements.

The STEPLIB concatenation of the ssnmDBM1 address space procedure includes a commented-out DD for the IBM Language Environment data sets SCEERUN and SCEERUN2. If your system does not include SCEERUN and SCEERUN2 in the system link list, you must uncomment this DD.

2. Add the address space startup procedures for the Db2-supplied WLM environments to by tailoring and running job DSNTIJMW. Job DSNTIJMW adds the following procedures to the system PROCLIB for WLM environments:
   - ssnmWLM for the sample WLM environment
   - ssnmWLMC for DSNWLM_DSNACICS
3. Copy the Db2 language procedures by running job DSNTIJMP. Job DSNTIJMP adds the following Db2-supplied language procedures to the target system procedure library:

- ssnmWLMD for DSNWLM_DEBUGGER
- ssnmWLMG for DSNWLM_GENERAL
- ssnmWLMJ for DSNWLM_JAVA
- ssnmWLMM for DSNWLM_MQSERIES
- ssnmWLM for DSNWLM_PGM_CONTROL
- ssnmWLMR for DSNWLM_REXX
- ssnmWLM1 for DSNWLM_NUMTCB1
- ssnmWLMU for DSNWLM_UTILS
- ssnmWLMW for DSNWLM_WEBSERVICES
- ssnmWLMX for DSNWLM_XML

Change any data set names that differ at your site. If you specified a data set name suffix in the DATASET NAME SUFFIX field of DSNTIPA1, that suffix is appended to data sets &USER..DBRMLIB.DATA, &USER..RUNLIB.LOAD, and &USER..SRCLIB.DATA. To override these data set names, you must edit the updates to the target system procedure library.

The language preparation procedures in job DSNTIJMP use the DISP=OLD parameter to enforce data integrity. However, when the installation CLIST runs, the DISP=OLD parameter for the DBRM library data set is modified to DISP=SHR. This might cause data integrity problems when you run multiple precompiler jobs. To avoid these data integrity problems, if you are not using DFSMSdfp partitioned data set extended (PDSE), you must change the language preparation procedures (DSNHICOB, DSNHCIB2, DSNHFOR, DSNHC, DSNHCPP, DSNHCPP2, DSNHPLI, DSNHASM, and DSNHSQL) to specify the DISP=OLD parameter instead of the DISP=SHR parameter.

If compiler STEPLIB statements are needed, add them.

4. Create environment files for the Db2-supplied Java WLM environment by tailoring and running job DSNTIJMJ. Job DSNTIJMJ has the following steps that create environment files for the Db2-supplied WLM environment, DSNWLM_JAVA:

a. Job step DSNTIJJ creates a Language Environment run-time options (JAVAENV) data set for the ssnmWLMJ address space procedure. This data set is allocated by the JAVAENV DD of the ssnmWLMJ address space procedure that is created in job DSNTIJMW.

In this job step, review the JAVA_HOME path, which is the path where the JDK is located, and update if necessary. If you are using a 64-bit JVM, use the following path:

JAVA_HOME=/usr/lpp/java160/J6.0_64
If you are using a 31-bit JVM, use the following path for Java 5:

```
JAVA_HOME=/usr/lpp/java150/J5.0
```

Or use the following path for Java 6:

```
JAVA_HOME=/usr/lpp/java160/J6.0
```

You cannot specify two JDKs in a single `JAVA_HOME` path.

- `_CEE_ENVFILE` if present, must be set to the name and path of the environment variable file that was created by job step `DSNTIJR`.
- `JVMPROPS` needs to be set to the name and path of the JVM startup options file that was created by job step `DSNTIJS`.
- `DB2_BASE` and `JCC_HOME` need to be set to the names of the HFS directories that you selected for the Db2 for z/OS and JCC product classes.

You cannot specify two JDKs in a single `JAVA_HOME` path.

b. Job step `DSNTIJR` creates an environment variable file in HFS.

c. Job step `DSNTIJS` creates a JVM startup options file in HFS.

d. The absolute path and name of a HFS or zFS file is passed to `DSNTJMJX`.

e. `DSNTJMJX` checks whether the directory where this file is located exists or not. If the directory does not exist, it is created.

5. After running job `DSNTIJMJ`, review the content of the Language Environment options data set and adjust the following environment variables as needed. The data set is created by `DSNTIJMJ` and must be allocated by the `JAVAENV DD` of the address space procedure for the Db2 core WLM environment for Java routines.

- If `_CEE_ENVFILE` is present, set it to the name and path of the environment variable file that was created by job step `DSNTIJR`.
- Set `JVMPROPS` the name and path of the JVM startup options file that was created by job step `DSNTIJS`.
- Set `DB2_BASE` and `JCC_HOME` to the names of the HFS directories that you selected for the Db2 for z/OS and JCC product classes.

### Installation step 4: Define the ICF catalog and alias: DSNTIJCA (optional)

The ICF catalog is the VSAM object in which Db2 records the data sets that you create during installation. Db2 uses the catalog alias as the prefix for your Db2 VSAM data sets.

#### About this task

Creating the catalog is optional. If you specified YES for the DEFINE CATALOG option on installation panel `DSNTIPA2`, you must create the catalog. Do not run this job if you want to use an existing ICF catalog and alias (that is, you specified NO for the DEFINE CATALOG parameter on installation panel `DSNTIPA2`). However, ensure that the ICF catalog that you are going to use is created and that you defined an ICF catalog alias.

#### Procedure

To create the ICF catalog and its alias:
1. Before running job DSNTIJCA, examine the DEFINE UCAT statement carefully to ensure that the parameters are appropriate for your needs.

2. Run job DSNTIJCA.
   If DSNTIJCA runs successfully, it produces return codes of 0.

3. If DSNTIJCA fails or abends, delete the ICF catalog (if it was created) and run the job again.

**Related tasks:**

- Installation step 1: Define the SMS environment for Db2 catalog and directory data sets: DSNTIJSS

**Related reference:**

- DEFINE CATALOG field

---

**Installation step 5: Define system data sets: DSNTIJIN**

Use job DSNTIJIN to define the VSAM and non-VSAM data sets for Db2.

**Before you begin**

Before you define the system data sets, ensure that you have defined an SMS environment for data sets that are associated with the Db2 catalog and directory. Data sets for the Db2 catalog and directory are managed by Db2 and must be SMS-managed. The SMS environment that you use for Db2 catalog and directory data sets must be established before you begin installation or migration to Db2 12. The SMS environment must include a data class for allocating the data sets in extended format and using extended addressability.

**Procedure**

To define system data sets:

1. Check the DEFINE CLUSTER statements in job DSNTIJIN to ensure that they allocate adequate disk space for your system.
   
   **Recommendation:** For recovery purposes, place system data sets such as the Db2 recovery log and the VSAM catalog on different disk volumes. Because these data sets are used frequently, do not migrate them by using DFSMSHsm.

2. Run job DSNTIJIN.
   If DSNTIJIN runs successfully, it produces return codes of 0 for all DEFINE statements and steps. Check any VSAM messages carefully.

3. If job DSNTIJIN fails or abends, complete the following steps:
   a. Remove the z/OS catalog delete statements from job DSNTIJDE.
   b. Run job DSNTIJDE. This job deletes the previously created data sets for the Db2 directory and Db2 catalog.
   c. Run DSNTIJIN again.

**Related concepts:**

- Allocating and extending data sets (Db2 Performance)

**Related tasks:**

- Installation step 1: Define the SMS environment for Db2 catalog and directory data sets: DSNTIJSS
**Job DSNTIJIN**

During installation, job DSNTIJIN defines VSAM and non-VSAM data sets for Db2.

DSNTIJIN performs the following tasks:

- Defines three non-VSAM data sets for the Db2 sample objects:
  - prefix.DBRMLIB.DATA
  - prefix.SRCLIB.DATA
  - prefix.RUNLIB.LOAD
- Defines the VSAM clusters for the bootstrap data sets.
  Each bootstrap data set (BSDS) consists of a VSAM key-sequenced data set. You defined the BSDS names during the ISPF tailoring session.
- Defines the VSAM clusters for the active log data sets.
  You specified up to 93 primary active log data sets during the ISPF tailoring session (NUMBER OF LOGS field on installation panel DSNTIPL). You might also have requested dual logging to generate two copies of each active log data set. Consequently, job DSNTIJIN can define up to 186 active log data sets. If you specified that DSNTIJIN is to create fewer active log data sets, then after installation, you can run the DSNJU003 utility to add additional active log data sets to the BSDS.
- Defines the VSAM clusters for the task list data set of the administrative task scheduler.
- Defines the VSAM clusters for the Db2 directory.
- Defines the VSAM clusters for the Db2 catalog.
- Invokes the LISTCAT command of access method service so that you can verify that the VSAM definitions were successful.
- Creates control intervals based on the value you specified in VARY DS CONTROL INTERVAL on panel DSNTIP7. If you specified YES, Db2 creates 4-KB, 8-KB, 16-KB, and 32-KB control intervals as appropriate in the AMS DEFINE CLUSTER commands for the Db2 catalog and directory data sets. If you specified NO, these data sets are created using a fixed control interval of 4-KB.

**Job DSNTIJDE**

Job DSNTIJDE deletes the Db2 system data sets that are defined by job DSNTIJIN. If a job fails or abends, you might need to run DSNTIJDE job before restarting the Db2 installation process.

Job DSNTIJDE is not part of the normal installation process; use this job only for rerunning part of the process. Do not run this job during migration or fallback.

In most cases, you must remove or comment out the delete statement in this job for the ICF catalog (if the statement is present). The ICF catalog probably does not need to be deleted and redefined.

Deletes might fail for data sets that do not exist. This does not necessarily indicate that the job failed. If you receive other messages, check them carefully.

Job DSNTIJDE does not work properly if job DSNTIJSG or job DSNTIJTM has been executed. Job DSNTIJDE does not delete the resource limit specification table, the data sets that are used by the distributed data facility, or the data sets that are created for work file table spaces.
If job DSNTIJDE fails or abends, correct the error conditions and rerun the job. If you want to delete the ICF catalog, first list its contents and delete the data sets that are cataloged there. This could include sample data sets, user-defined data sets, or subsystem data sets that were not deleted properly. You can use a FORCE command to delete the user catalog. If you delete the catalog using FORCE before deleting all the data sets, you can use the RECATALOG option of DEFINE CLUSTER and delete the data sets.

**Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL**

For Db2 12 installation, you must define initialization parameters in the subsystem parameter module (DSNZPxxx) and application defaults load module (dshideep), define the offline message generator CCSID (DSNHMCID), and update DDF related information in the BSDS.

**About this task**

The subsystem parameter module (DSNZPxxx) and application defaults module (dshideep) names are specified in the following DSNITPO3 installation panel fields:

- **PARAMETER MODULE field**
- **APPL DEFAULTS field**

**Procedure**

To define the Db2 12 initialization parameters, complete the following steps:

1. Run job DSNTIJUZ. Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

   You might need to make the following adjustments before running the job:
   - If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLOAD, you can move the SYSLMOD output to that library.
   - If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
   - If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This action ensures that members DSNFSTMP, DSNJARVP, DSNJLOGP, DSNTSPRM, DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSREFIX:

   ```
   SET BOUNDARY (GLOBAL).
   LIST DDDEF (SMPTLIB).
   ```

   Insert the DSREFIX value after SDSNLOAD and ADSNLOAD.

   When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions.

   For more information, see [job DSNTIJUZ: define the Db2 data-only subsystem parameter module](#)
2. Run job DSNTIJUA. Job DSNTIJUA defines the Db2 data-only application defaults module.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of `prefix:SDSNEXIT` and `prefix:SDSNLOAD`, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUA to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUA so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZQ. This action ensures that member DSNARIB is linked with the hardware application defaults module. SMPTLIB is `hlq.HDBCC10.F2`, where `hlq` is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:
  
  ```
  SET BOUNDARY (GLOBAL);
  LIST DDEFF (SMPTLIB);
  ```

  Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

For more information, see [Job DSNTIJUA: define data-only application defaults module](#).

3. Run job DSNTIJUM. Job DSNTIJUM defines the Db2 data-only offline message generator CCSID (DSNHMCID) module which is needed for message conversion by Db2 applications and utilities. For more information, see [Job DSNTIJUM: define Db2 data-only offline message generator CCSID (DSNHMCID) module](#).

4. Run job DSNTIJUL. Job DSNTIJUL runs the Db2 change log inventory (DSNJU003) to update DDF-related information in the BSDS. For more information, see [Job DSNTIJUL: update DDF related BSDS information](#).

## Installation step 7: Initialize system data sets: DSNTIJID, DSNTIJIE, and DSNTIJIF

After you define your system data sets and Db2 initialization parameters, you must initialize your system data sets.

### Procedure

To initialize system data sets:

Run the following jobs in the specified order:

1. DSNTIJID
2. DSNTIJIE
3. DSNTIJIF

- If the jobs run successfully, they produce return codes of 0. Otherwise, check for and resolve error conditions that might be reported by VSAM messages.

Chapter 5. Installing Db2 425
• If any of the jobs fail or abend, remove the ICF catalog delete statements (job step DSNTIVC) from job DSNTIJDE, run job DSNTIJDE, and then rerun the following jobs:
  – DSNTIJIN
  – DSNTIJUZ
  – DSNTIJUA
  – DSNTIJUM
  – DSNTIJUL
  – DSNTIJID
  – DSNTIJIE
  – DSNTIJIF

Results

Job DSNTIJID records the active log data set names to the BSDS, formats the active log data sets, and initializes the Db2 directory table spaces and indexes.

Job DSNTIJIE initializes the Db2 catalog table spaces and indexes (through the SYSGRTNS table space).

Job DSNTIJIF initializes the remaining Db2 catalog table spaces and indexes.

Related concepts:
Installation step 8: Define user authorization exit routines: DSNTIJEX (optional)

Installation step 8: Define user authorization exit routines: DSNTIJEX (optional)

Job DSNTIJEX builds sample authorization exit routines from the source code in prefix.SDSNSAMP. These sample authorization exit routines are DSN3@SGN and DSN3@ATH, and the user version of the access control authorization exit routine, DSNX@XAC.

Job DSNTIJEX includes a step to assemble and link-edit the sample version of DSNACICX, which you can use to modify CICS parameters that the DSNACICS caller specifies. Then DSNTIJEX places the exit routines in the prefix.SDSNEXIT library. The Db2 CLIST tailors the JCL in DSNTIJEX to match your site’s environment.

The sample authorization exit routines are not the same as the default authorization exit routines that are supplied by Db2. By implementing the sample authorization exit routines, you can provide group names as secondary authorization IDs. By modifying the sample authorization exit routines, you can tailor authorization processing for your subsystem.

DSNXXSXAC is a copy of the default access control authorization exit routine that users can modify. This exit routine allows you to bypass some or most of Db2 authorization checking to specify your own authorization checking. If you do not modify it, this step is not needed and you should delete it.

DSNACICS is a stored procedure that invokes user exit routine DSNACICX, which you can use to modify CICS parameters that the DSNACICS caller specifies. If you
do not need to modify the caller’s parameter values, you can use the default DSNACICX exit routine. However, if you need to modify the caller’s parameter values, you need to perform the following tasks:

1. Write a user exit routine in assembler, COBOL, C, or PL/I
2. Assemble or compile the source code
3. Link-edit the object code into the Db2 exit routine library

Installation job DSNTIJEX includes a step to assemble and link-edit the sample version of DSNACICX. You can use this step as a model for your program preparation job.

You have the following options regarding exit routines:

- To use the sample authorization exit routines, run job DSNTIJEX.
- To use the default authorization exit routines, skip job DSNTIJEX.
- To use the modified sample authorization exit routines, modify DSNTIJEX to reference the correct library before you run it.

If you will use the RACF/Db2 external security module (DSNXRXAC) as your Db2 access control authorization exit routine, modify DSNTIJEX to refer to DSNXRXAC instead of DSNXSXAC.

If job DSNTIJEX runs successfully, it produces return codes of 4.

If job DSNTIJEX fails or abends, correct the problem and rerun the job.

Related concepts:
- [Db2 data access control (Managing Security)]
- [The default Db2 exit routine (RACF Access Control Module Guide)]

Related reference:
- [DSNTIPO3: Default startup modules panel]

Related information:
- [Exit routines (Db2 Administration Guide)]

### Installation step 9: Record Db2 data to SMF (optional)

When you install Db2, you can specify whether Db2 statistical, accounting, and audit trace data is to be collected.

**Procedure**

To specify the data that is to be collected:

1. Choose from the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To have Db2 collect statistical information</td>
<td>Accept the default (YES class 1) for the SMF STATISTICS option on installation panel DSNTIPN. To collect statistical information for deadlock or timeout, specify class 3. To collect information about DDF error conditions, specify class 4.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>To have Db2 collect accounting information</td>
<td>Accept the default (1) or specify * (all classes) for the SMF ACCOUNTING option on installation panel DSNTIPN.</td>
</tr>
<tr>
<td>To have Db2 collect auditing information</td>
<td>Specify * (all classes) for the AUDIT TRACE option on installation panel DSNTIPN.</td>
</tr>
</tbody>
</table>

In all cases, Db2 invokes a trace, passing the data it collects to the System Management Facility (SMF) of z/OS.

Db2 also passes performance data to SMF whenever an accounting, statistics, or audit trace is successfully started or stopped. Db2 can also record other performance data. After you complete the installation process, you can use commands to have Db2 record performance data for subsystem events.

2. If, during installation, you requested that Db2 pass accounting and statistics data to SMF, update the SMFPARMxx member of SYS1.PARMLIB as follows:
   - Specify the ACTIVE parameter.
   - Specify STC,NODETAIL for the SUBSYS parameter.
   - Specify the proper TYPE subparameter of the SYS and SUBSYS parameters.

**What to do next**

During Db2 execution, you can use the SMF SET or SS command to alter the SMF parameters. For example, you can record the statistics trace class 1 IFCIDs 0001, 0002, and 0202 (SMF record type 100); accounting trace class 1 IFCIDs 0003 and 0239 (SMF record type 101); and all other Db2 trace records (SMF record type 102) to SMF. To record this information, issue the following command:

```
SYS(TYPE(100:102))
```

For Db2 to pass data to SMF, you must allocate an adequate supply of SMF buffers. The default buffer settings are probably insufficient.

You can specify SMF buffering on the VSAM BUFSP parameter of the Access Method Services DEFINE CLUSTER statement. Do not use the default settings if Db2 data is sent to SMF. Specify CISZ(4096) and BUFSP(81920) on the DEFINE CLUSTER statement for each SMF VSAM data set. These values for CISZ and BUFSP are the minimum requirement for Db2. You might need higher values for CISZ and BUFSP, depending on the requirements of all your z/OS subsystems.

You can also code an IEFU84 SMF exit routine to process the records that are produced.

**Related tasks:**
- [Controlling traces (Db2 Administration Guide)](#)

**Related reference:**
- [START TRACE (Db2) (Db2 Commands)](#)
- [z/OS MVS Initialization and Tuning Guide](#)

**Related information:**
- [z/OS MVS Initialization and Tuning Reference](#)
Installation step 10: Establish subsystem security (optional)

Db2 can control access to data within a Db2 subsystem. Db2 also works with outside security systems, such as RACF, that control access to the Db2 subsystem.

Before you begin

Your system security administrator should have provided a Db2 for z/OS security environment as outlined in the information about managing Db2 security, including user IDs and resource profiles for batch connections. If you are using RACF, ensure that the system security administrator also defined your Db2 resources to RACF.

The following IDs should have been specified on installation panel DSNTIPPI:

- Two system administrator (installation SYSADM) authorization IDs
- Two system operator (installation SYSOPR) authorization IDs
- Two security administrator (installation SECADM) authorization IDs or roles
- One authorization ID (installation IBMUSER) if RACF is not available for batch access and USER= is not specified in the JOB statement

Procedure

To establish subsystem security:

- Ensure that the specified installation SYSADM IDs are defined in your TSO and RACF systems before attempting to access Db2. You can also define installation SYSOPR IDs, SECADM IDs, and the installation IBMUSER ID.
- To enable primary and secondary user IDs to issue Db2 commands from the z/OS console or TSO SDSF, define RACF classes to authorize Db2 commands. Use the following RACF commands:

  ```
  SETR CLASSACT(DSNADM)
  RDEFINE DSNADM DSN1.SYSOPR UACC(NONE)
  SETR RACLIST(DSNADM) REFRESH
  PERMIT DSN1.SYSOPR CLASS(DSNADM) ID(userid) ACCESS(READ)
  SETR RACLIST(DSNADM) REFRESH
  ```

  You can grant SYSOPR authority to all primary and secondary user IDs that issue Db2 commands from the z/OS console or TSO SDSF, but only after Db2 can process SQL.

- To separate security tasks from system administration tasks, use the SEPARATE SECURITY field on the DSNTIPPI panel. Before setting the SEPARATE SECURITY field to YES, set at least one SECADM system parameter to an authorization ID, or create the necessary trusted contexts and roles. If you specify YES for SEPARATE SECURITY, system administrator authority cannot be used to perform security tasks, and the SECADM authority is required to manage trusted contexts and roles. If both SECADM system parameters are set to roles and those roles have not been created, no one will have the authority to manage trusted contexts and roles.

- Run job DSNTIJRA to complete the following tasks:
  - Define the administrative task scheduler started task module to RACF program control
  - Define the administrative task scheduler as a trusted context in RACF

    If you are using a Db2 data sharing environment, customize and run DSNTIJRA for each member of the group.

Related concepts:
Installation step 11: Connect Db2 to TSO

Although you can eventually connect Db2 to IMS, CICS, or both, connecting initially only to TSO is recommended for a first-time installation.

About this task

For a first-time installation, you can run the sample applications that do not require CICS or IMS, allowing your database and system administrators to gain familiarity with the administrative facilities of this version of Db2.

If you have previously installed Db2 and are performing that task again, your database and system administrators are probably already familiar with Db2. In this case, you can connect IMS, CICS, or both, at the same time that you connect TSO. You can then run the sample applications that require CICS and IMS at the same time that you run the sample applications for TSO and batch.

Procedure

To attach Db2 to TSO:
1. Make Db2 load modules available to TSO and batch users.
2. Make Db2 CLISTs available to TSO and batch users.
3. Make PL/I options available (if applicable).
4. Make panels, messages, and load modules available to ISPF and TSO.
5. Connect the DB2I panels to the ISPF Main Panel.

Make Db2 load modules available to TSO and batch users

As part of installation, you must make Db2 load modules available to TSO and batch users. If prefix.SDSNEXIT and prefix.SDSNLOAD are not included in your LNKLISTxx, you can skip this step.

Procedure

To make Db2 load modules available to TSO and batch users:

If you have not included prefix.SDSNEXIT and prefix.SDSNLOAD in your LNKLISTxx, you must add STEPLIB statements to your logon procedures and to the JCL for jobs to ensure that you access the Db2 12 load modules. If prefix.SDSNEXIT is not in your LINKxx, add it to your STEPLIB and JOBLIB concatenations before prefix.SDSNLOAD.

Related tasks:
Choosing link list options

Make Db2 CLISTs available to TSO and batch users (DSNTIJVC)

To connect Db2 to TSO, you must make Db2 CLISTs available to TSO and batch users by running the DSNTIJVC job.
About this task

From `prefix.SDSNCLST`, the CLIST reads and edits these four CLISTS: DSNEMC01, DSNH, DSNU, and DSNHC. It then places those CLISTS in `prefix.NEW.SDSNTEMP`. You might want to modify the default values. The DSNEMC01 CLIST provides installation default values for the DB2I Default panels (option D on panel DSNEPRI). The first time that a TSO user executes DB2I on a specific ISPF application such as DSNE (NEWAPPL), the DSNEMC01 CLIST sets the defaults based on the values that are specified on installation panel DSNTIPF. DSNEMC01 stores the values in the ISPF profile member DSNEPROF.

Job DSNTIJVC merges the tailored CLISTS from `prefix.NEW.SDSNTEMP` with unchanged CLISTS and REXX execs from `prefix.SDSNCLST`. Then job DSNTIJVC places all CLISTS and REXX execs in `prefix.NEW.SDSNCLST`. It also converts the record format of the Db2 CLISTS from fixed-block to variable-block format with a record length of 84 and a block size of 3120. A CLIST that has been converted from fixed-block to variable-block format cannot be used as input to the DSNTINST CLIST; use the unedited version of the SDSNCLST data set, as created by SMP/E.

Procedure

To make the CLISTS available to TSO and batch users:

1. If you use fixed-block format for your CLIST libraries, modify job DSNTIJVC as follows:
   - Change the SYSIN DD statement to DUMMY.
   - Change the allocation of `prefix.NEW.SDSNCLST` to match the data control block (DCB) attributes of your other CLIST libraries.
2. Run job DSNTIJVC.
3. Either concatenate `prefix.NEW.SDSNCLST` with your existing CLIST libraries or copy `prefix.NEW.SDSNCLST` into an existing CLIST library.

What to do next

If you need to rerun job DSNTIJVC, first delete data set `prefix.NEW.SDSNCLST`, which is created by this job.

When corrective service is applied to a CLIST, SMP/E changes only the `prefix.SDSNCLST` data set. You need to redo any record format changes and reapply any special tailoring that is required. You also need to move the CLIST to `prefix.NEW.SDSNCLST`. Corrective service (program temporary fixes) for these CLISTS is sent with ++HOLD statements, calling to your attention the possibility of additional work.

Related tasks:

- Completing the CLIST processing

Ensure that PL/I options are available

If you are using PL/I, ensure that the options that your Db2 programmers use are included in the compiler.

Procedure

To ensure that PL/I options are included in the compiler:
1. Ensure that the following options that are used by the program preparation function are available:

FLAG   OBJECT   SOURCE   TERMINAL   XREF

Restrictions that are imposed by your site on PL/I compiler options affect how you can use Db2 program preparation.

2. If the macro pass is used, ensure that the following options are also available:

MACRO   MDECK   SYNTAX

Make DB2I panels, messages, and load modules available to ISPF and TSO

During installation, to connect the Db2 Interactive facility (DB2I) to TSO, you must make panels, messages, and load modules available to ISPF and TSO.

About this task

DB2I uses the ISPF PROFILE and SHARED variable pools for most panel variable fields. As a result, you can easily re-enter a panel when panel variables have previously been specified. For the Db2 subcommands that permit lists of plan names, package names, DBRMs, and ENABLE and DISABLE statements, DB2I provides ISPF to contain all the user-specified variables for these subcommand keywords.

DB2I creates and maintains a set of ISPF tables in a user-defined TSO data set that is allocated to a data set with a ddname of DSNETBLS. The DB2I-generated tables in this library are DSNCONNS, DSNDBRMS, and DSNPLPKN. The following table shows the library table member names and their contents.

<table>
<thead>
<tr>
<th>Table 43. The Db2 ISPF table library</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNCONNS</td>
</tr>
<tr>
<td>DSNDBRMS</td>
</tr>
<tr>
<td>DSNPLPKN</td>
</tr>
<tr>
<td>DSNPATHS</td>
</tr>
</tbody>
</table>

DB2I uses ISPF table services to maintain individual ISPF tables within the DSNETBLS data set. For performance reasons, ISPF keeps this table library in an open state once an individual table has been updated. Attempts to close this data set by using the TSO FREE command results in error message IKJ56861I.

Procedure

1. If you added the DB2I libraries as part of Tailoring Db2 12 installation and migration jobs with the CLIST you can skip this step. Otherwise, concatenate the Db2 ISPF libraries with the ISPPLIB, ISPTLIB, ISPSLIB, and ISPMLIB DD statements in your logon procedures. You must also concatenate the Db2 ISPF libraries with these DD statements in any of your CLISTS where the libraries might be allocated. These libraries are prefix.SDSNSPFP, prefix.SDSNSSPFM, prefix.SDSNSPSF, and prefix.SDSNSPFPE or prefix.SDSNPFP. If you use English
DB2I panels, concatenate SDSNPFPFE to ISPPLIB. If you use Kanji DB2I panels, concatenate SDSNPFPK instead of SDSNPFPFE.

PROFILE WTP MSGIO
ALLOCATE DDNAME(SYSPROC) DSN('prefix.SDSNCLST') SHR REUSE
ALLOCATE DDNAME(ISPMLIB) DSN('prefix.SDSNSPFM' + 'ISP.SISPMENU') SHR REUSE
ALLOCATE DDNAME(ISPPLIB) DSN('prefix.SDSNSPFM' + 'prefix.SDSNPFP' + 'ISP.SISPENU') SHR REUSE
ALLOCATE DDNAME(ISPPLIB) DSN('prefix.SDSNSPF' + 'prefix.SDSNPFP' + 'prefix.SDSNPFP' + 'ISP.SISPENU') SHR REUSE

2. Create the DSNETBLS data set.
   When allocating this data set, you should assign the following DCB attributes, where n is any integer:
   DSORG(PO) RECFM(FB) LRECL(80) BLKSIZE(n*LRECL)
   The following example shows how to set up an ALLOCATE statement to create the data set:
   ALLOC DA(DSNPFT) NEW SP(11) TR DIR(10) +
   DSORG(PO) RECFM(FB) LRECL(80) BLKSIZE(800)
   F(DSNETBLS) REUSE
   The following example shows how to allocate an existing data set to the data set with the DSNETBLS ddname:
   ALLOC DA(DSNPFT) F(DSNETBLS) REUSE

3. If you want to run the ISPF/CAF sample application that is provided with Db2, ensure that the data set prefix.RUNLIB.LOAD is included in the logon procedures or in the ISPLLIB concatenation. If you chose IBMCOB for your LANGUAGE DEFAULT on panel DSNTIPF, you should also include the data set prefix.CEE.SCEERUN.

Related concepts:
Dynamic SQL and the ISPF/CAF application

Related reference:
[1] z/OS ISPF Messages and Codes

Connect DB2I panels to the ISPF main panel
You can connect the DB2I panels to the standard ISPF panels that are already installed on your system.

About this task

Recommendation: Use the following panels for establishing the connection. See your TSO administrator for other possibilities.

- Use panel ISP@MSTR, ISR@PRIM, or ISRFP A for the connection to Db2 Interactive services.
- Use panel ISR00003 for the tutorial menu update.

Two example panels are provided. Their names are DSN8ISPM (the Db2 version of an ISPF primary options panel) and DSN8ISPT (the Db2 version of a tutorial table of contents).

If the Db2 panel library is concatenated before the standard ISPF library, the connection is made.
Procedure

To connect DB2I panels to the ISPF main panel:

1. Using the TSO RENAME command, give DSN8ISPM an alias of ISR@PRIM, and give DSN8ISPT an alias of ISR00003. For example:

```
RENAME 'prefix.SDSNSPFP(DSN8ISPM)' (ISR@PRIM) ALIAS
RENAME 'prefix.SDSNSPFP(DSN8ISPT)' (ISR00003) ALIAS
```

2. Modify the panels as shown in the following examples. If your site has made changes to either of these panels, change your existing panels rather than using the examples.

Examples

```
)ATTR
/**********************************************************************/
/* COPYRIGHT = 5740-XYR (C) COPYRIGHT IBM CORP 1982, 1985, 1990, 2003 */
/* REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083 */
/* STATUS = VERSION 8, LEVEL 0 */
/**********************************************************************/
)BODY
%----------------------- ISPF/PDF PRIMARY OPTION MENU ------------------------
%OPTION ===> ZCMD +USERID - &ZUSER.
% 0 +ISPF PARMS - Specify terminal and user parameters +TIME - &ZTIME.
% 1 +BROWSE - Display source data or output listings +TERMINAL - &ZTERM.
% 2 +EDIT - Create or change source data +PF KEYS - &ZKEYS.
% 3 +UTILITIES - Perform utility functions
% 4 +FOREGROUND - Invoke language processors in foreground
% 5 +BATCH - Submit job for language processing
% 6 +COMMAND - Enter TSO command or CLIST
% 7 +DIAGL0 TEST - Perform dialog testing
% 8 +DB2I - Perform DATABASE 2 Interactive functions
% C +CHANGES - Display summary of changes for this release
% 8 +EXIT - Terminate ISPF using log and list defaults
% Enter%END+command to terminate ISPF.
%
)INIT
.HELP = ISR00003
&ZPRIM = YES /* ALWAYS A PRIMARY OPTION MENU */
&ZHTOP = ISR00003 /* TUTORIAL TABLE OF CONTENTS */
&ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE */
)PROC
&ZSEL = TRANS( TRUNC (&ZCMD'.')
0,'PANEL(ISPOPTA)'
1,'PGM(ISRBRO)'
2,'PGM(ISREDIT)'
3,'PANEL(ISRUTIL)'
4,'PANEL(ISRFPA)'
5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
6,'PGM(ISRPTC)'
7,'PGM(ISPYXDR) PARM(&ZTAPPLID) SCRNAME(DTEST) NOCHECK'
8,'CMD(DSNECPRI) NEWAPPL(DSNE)'
C,'PGM(ISPTUTOR) PARM(ISR00005)'
T,'PGM(ISPTUTOR) PARM(ISR00000)'
X,'EXIT'
*/?')
&ZTRAIL = .TRAIL
)END
```

Figure 78. ISPF primary option panel (DSN8ISPM), edited to include Db2I
The figure above shows panel DSN8ISPM. Notice the added lines in boldface type. Adding these lines allows you to invoke the Db2 Interactive (Db2I) functions. The added lines include one displayed line:

```
% 8 +DB2I   - Perform DATABASE 2 Interactive functions
```

You can also choose to add one of these undisplayed lines:
```
8,'CMD(DSNECPRI) NEWAPPL(DSNE)'
8,'CMD(DSNECPRI SSID(xxxx)) NEWAPPL(DSNE)'
```

The displayed line lets the user choose DB2I. Both of the undisplayed lines invoke the Db2I main panel (DSNEPRI). If you use the first undisplayed line, you accept the default for the subsystem identifier (SSID) parameter. If you use the second undisplayed lines, you can specify a different SSID parameter.

DSNECPRI is a CLIST and can be invoked directly from another user CLIST. It is an alternative way to invoke DSNEPRI without updating the primary ISPF panel.

By specifying NEWAPPL(DSNE), you define DSNE as the ISPF application that DB2I uses. ISPF uses the name DSNE to create the ISPF profile pool member name (DSNEPROF) in the TSO_userid.ISPPROF data set, which contains all ISPF panel variables defined during DB2I execution. Any customized DSNEPROF members can be migrated from Db2 12.

**Recommendation:** Examine any new or changed default panel values to ensure that your custom values are still valid, specifically the option values for the subcommands BIND PLAN, REBIND PLAN, BIND PACKAGE, and REBIND PACKAGE.

**Using a NEWAPPL name other than DSNE:** You can define any valid ISPF application name.
The figure above shows panel DSN8ISPT. Notice the two added lines in boldface type. Adding these lines allows you to invoke the Db2 tutorial panels. One of the two added lines is displayed:

%8+ DB2 - Information about DB2

The other added line is undisplayed:

8,DSN4V2DB

The displayed line presents the user with a choice for the Db2 tutorial. The undisplayed line actually invokes the Db2 tutorial menu (DSN4V2DB).

Related reference:
Installation step 12: Connect Db2 to IMS (optional)

Connecting Db2 to IMS is an optional part of installation. Connecting Db2 to IMS requires coordination with your company’s group that supports IMS systems.

Procedure

To connect the IMS attachment facility:
1. Make Db2 load modules available to IMS
2. Define Db2 to IMS
3. Define new programs and transactions to IMS
4. Prepare IMS applications for Db2
5. Depending on your site, you might also need to:
   • Define Db2 plans for IMS applications
   • Generate a user language interface

Related tasks:
Connecting the IMS attachment facility

Installation step 13: Connect CICS to Db2 (optional)

To connect Db2 to CICS, you need to disconnect and connect the CICS region.

About this task

Coordinate this connection with your CICS support group.

Related tasks:
Defining the CICS Db2 connection

Installation step 14: Define Db2 to z/OS

After you run job DSNTIJMV to define Db2 in SYS1.PARMLIB, you must activate the definition in z/OS.

About this task

Job DSNTIJMV makes the following changes to the SYS1.PARMLIB library:
• Creates new subsystem definitions in the IEFSSNxx member
• Creates new APF libraries in the IEAAPFx member
• Creates new load module libraries in the LNKLSTxx member

The load module library SDSNLINK contains the early code. SDSNLINK contains modules that must be placed in the link list look-aside address space (LLA) because they are loaded at subsystem initialization during the IPL.

Procedure

To define Db2 to z/OS:
IPL z/OS.
During the z/OS IPL, message DSN3100I appears on the z/OS console, stating that
Db2 is ready for the START command.
Alternatively, to define Db2 to z/OS until the next IPL, you can use these
commands, in the following order:
1. SETPROG LNKLST, to update the link list concatenations
2. MODIFY LLA,REFRESH, to rebuild the LLA directory for the entire set of
libraries that are managed by the LLA. This action is also known as a complete
refresh of the LLA.
3. SETPROG APF, to update the APF list
4. SETSSI ADD, to dynamically add the new Db2 and IRLM subsystems

Example: Suppose that you want to perform these actions to define a Db2
subsystem to z/OS before the next IPL:
1. Add the Db2 ERLY code to a link list data set named MY.LNKLST.
2. Perform a complete refresh of the LLA.
3. Add the Db2 and IRLM libraries to the dynamic APF list.
4. Dynamically add a new Db2 subsystem named DSN1, with these
characteristics:
   • Command prefix -DSN
   • Command prefix scope S (started)

Use the following commands:

```
SETPROG LNKLST,ADD,NAME=MY.LNKLST,DSNAME=prefix.SDNSLNK
MODIFY LLA,REFRESH
SETPROG APF,FORMAT=DYNAMIC,ADD,DSNAME=prefix.SDNSLNK
SETPROG APF,FORMAT=DYNAMIC,ADD,DSNAME=prefix.SDNSEXIT
SETPROG APF,FORMAT=DYNAMIC,ADD,DSNAME=prefix.SDNSLOAD
SETPROG APF,FORMAT=DYNAMIC,ADD,DSNAME=prefix.SDXRRESL
SETSSI ADD,SUB=DSN1,INITRTN=DSN3INI,INITPARM='DSN3EPX,-DSN,S'
```

Related reference:

- [z/OS SETPROG Command (MVS System Commands)]
- [z/OS SETSSI Command (MVS System Commands)]

### Installation step 15: Start the Db2 subsystem

During installation of Db2, start the Db2 subsystem after you IPL z/OS.

**Procedure**

To start Db2:

1. Start the IRLM, if you have not requested that Db2 automatically start the
IRLM. You must start IRLM before you start Db2.

   Use the following command:

   **START irlimproc**

   where *irlimproc* is the name that you specified for the PROC NAME option on
IRLM Panel 1 (DSNTIPI).

   If you specified YES for the AUTO START option on IRLM Panel 1 (DSNTIPI),
Db2 starts the IRLM automatically.
2. Start Db2 from the z/OS console. The simplest form of the following command is:

```
-DSN1  START  DB2
```

where `-DSN1` is the subsystem command prefix that you defined for Db2. When you issue the START DB2 command with this command format, Db2 uses the subsystem parameter module that is specified in the `ssnmMSTR` start-up JCL procedure in SYS1.PROCLIB.

If you need to change the name of the subsystem parameter module, you can edit the `ssnmMSTR` JCL member in SYS1.PROCLIB. Alternatively, you can override the default subsystem parameter module name by using the PARM option, as follows:

```
-DSN1  START  DB2,PARM('DSNZPxxx')
```

If you have not run job DSNTIJRT, and the administrative task scheduler starts, the administrative task scheduler issues error message DSNA679I to report SQLCODE -805 until you run job DSNTIJRT. The administrative task scheduler starts if you specified an address space procedure name in the ADMIN SCHEDULER field on installation panel DSNTIPX.

If Db2 starts successfully, two to five address spaces also start. These address spaces are `ssnmMSTR` and `ssnmDBM1`, possibly `ssnmDIST`, possibly `ssnmADMT`, and `irlmproc`, where `ssnm` is the Db2 subsystem name, `ssnmADMT` is the Db2 administrative task scheduler procedure name, and `irlmproc` is the IRLM procedure name.

If Db2 starts successfully, the series of RESTART messages that you receive concludes with these two messages:

```
DSNR002I  RESTART COMPLETED
DSN9022I  DSNYASCP 'DSN1 START DB2' NORMAL COMPLETION
```

When you start Db2 12 for the first time, Db2 issues message DSNT501I with reason code 00C900A6. This message is expected. When you run job DSNTIJTC, which is required to complete the tailoring of the Db2 catalog, the cause of this message is corrected. Job DSNTIJTC is described in a later installation step.

After you start Db2, identify unusual conditions for databases with the command:

```
-DSN1  DISPLAY  DATABASE(*+)  SPACENAM(*+)  RESTRICT
```

If Db2 does not start successfully, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. Ensure that the DSNTIJID, DSNTIJIE, DSNTIJIF, DSNTIJIN, and DSNTIJUZ jobs ran correctly. Also, verify that the subsystem parameter member that you specified (or allowed by default) when you started Db2 is the one that is built by job DSNTIJUZ. Check the JCL for the Db2 startup procedure.

If you use distributed data facility, you must define VTAM before DDF can start. However, you do not need to have TCP/IP configured to start DDF. In addition, transactions such as those from DDF or CICS may fail because work files have not yet been defined.

3. Optional: Start TSO. After Db2 starts, it is available to TSO. If you want to use the TSO SUBMIT command to do housekeeping and installation verification, you must start TSO (if it is not already started).
Installation step 16: Tailor the Db2 catalog: DSNTIJTC

Job DSNTIJTC invokes the CATMAINT utility to tailor your Db2 12 catalog, which you must do to fully install Db2. DSNTIJTC performs site-specific tailoring of the catalog.

Before you begin

To execute DSNTIJTC, you must have the installation SYSADM or installation SYSOPR authority. The presence of the installation SYSOPR authority in the privilege set allows you to install or migrate Db2 without access to user objects.

Also, SMS must be active on the system where this Db2 subsystem is being installed. SMS must be configured to allocate all data sets for the Db2 catalog and directory in extended format and to use extended addressability.

About this task

DSNTIJTC contains the following job steps:

- Job step DSNTITC, which creates new catalog and directory objects, adds columns to existing catalog tables, and creates and updates indexes on the catalog tables to accommodate new Db2 12 objects. All IBM-supplied objects are created or updated sequentially during the execution of this step.
- Job step DSNTIRI, which executes only when job step DSNTITC terminates abnormally or with a return code of 8 or greater. DSNTIRI rebuilds catalog indexes that were placed in REBUILD-pending status during execution of DSNTITC.

Important: In a data sharing environment, do not run DSNTIJTC after installing non-originating members.

Procedure

To tailor the Db2 catalog:

Run job DSNTIJTC. You must ensure that the installation job runs on the same z/OS system on which the appropriate Db2 subsystem is running.

A status message, DSNU777I, is issued at several points to indicate progress. Diagnostic error messages are issued when CATMAINT processing fails. If a problem is found during the SQL processing phase of installation, message DSNU778I is issued. If non-supported functions are encountered, message DSNU776I is issued. All of these messages are written to the SYSPRINT data set. Because this is the first use of Db2, errors from earlier steps might be detected here.

Results

If you receive abend reason code 00C200EF, the catalog and directory data sets are not in a compliant SMS environment.

If you receive an abend reason code from the data manager (X'00C9'xxxx) or buffer manager (X'00C2'xxxx), carefully recheck jobs DSNTIJIN, DSNTIJID, DSNTIJIE, and DSNTIJIF.

Related reference:

DSNTIPY: Job editing panel
**Installation step 17: Activate Db2 12 function levels: DSNTIJAF**

**Before you begin**

By default, new Db2 12 installations are at function level 500. However, you can also activate a higher function level during the installation process.

You must use a privilege set that includes the installation SYSADM or installation SYSOPR authority. The installation SYSOPR authority enables you to install or migrate to Db2 12 without access to user objects.

**About this task**

Job DSNTIJAF is tailored based on the TARGET FUNCTION LEVEL value that you specify on panel DSNTIP00 of the installation CLIST.

**Procedure**

To activate the function level:

Run job DSNTIJAF.

**Related reference:**

- [ACTIVATE (Db2) (Db2 Commands)](#)
- [What’s new in Db2 12 function levels (Db2 for z/OS What’s New?)](#)
- [Function level 500 (activated at Db2 12 installation or after migration—October, 2016) (Db2 for z/OS What’s New?)](#)
- [DSNTIP00: Activate a function level panel](#)

**Installation step 18: Create default storage group, define temporary work files, and bind Db2 REXX Language Support: DSNTIJTM**

You must create the default storage group that is for user-defined Db2 tables that are not specifically assigned to a storage group. You must also define the database that is for temporary work files, and bind Db2 REXX Language Support.

**Procedure**

To create the default storage group, define temporary work files, and bind Db2 REXX Language Support:
Run job DSNTIJTM.
If job DSNTIJTM runs successfully, it produces the return codes that are shown in the following table:

Table 44. DSNTIJTM return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIAD</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIAB</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIAS</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTICR</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIRX</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIST</td>
<td>(none)</td>
<td>0000</td>
</tr>
</tbody>
</table>

Results

DSNTIJTM assembles, link-edits, binds, and runs DSNTIAD, a program that processes certain SQL statements dynamically. It also defines the initial buffer pool sizes that are specified on installation panels DSNTIP1 and DSNTIP2. When you use the ENABLE or MEMBER functions, the steps to prepare and bind the DSNTIAD program are unnecessary and are removed during editing.

DSNTIJTM also provides some cleanup to ease reruns.

The SYSDEFLT storage group is the default storage group, and it is used for your database, table space, and table definitions that are not related to a specific storage group. By default, it is also used for the work file database and for work file table spaces that are created during Db2 installation and migration. When DSNTIJTM creates the SYSDEFLT storage group, it completes the following grants:

• Grants use of the default buffer pool and storage group to PUBLIC or the authorization IDs that you specified in the INSTALL GRANTEE(S) field of panel DSNTIPG.

• Grants use of the SYSDEFLT storage group to PUBLIC or to the authorization IDs that you specified in the INSTALL GRANTEE(S) field of panel DSNTIPG. This access is necessary to give users the ability to implicitly create table spaces in the default database.

• Grants authority to create tables and table spaces in the default database to PUBLIC or to the authorization IDs that you specified in the INSTALL GRANTEE(S) field of panel DSNTIPG.

What to do next

For non-data-sharing installations, the work file database name is DSNDB07. For data sharing installations, the work file database name is the name that was specified in the WORK FILE DB field on installation panel DSNTIPK. After creating the work file database, the DSNTIJTM job creates the work file table spaces as specified on installation panel DSNTIP9.

You can increase the number of additional temporary work file table spaces by increasing the values in the work file table space fields on panel DSNTIP9, particularly if you expect a large amount of sorting at your site. Additional
temporary work file table spaces can improve Db2 performance by reducing device contention among applications. These additional work files also can be used for sorting indexes on large tables during index creation.

You can choose to have job DSNTIJTM create these additional table spaces, or you can create them after you run DSNTIJTM. To create additional work file table spaces after you run job DSNTIJTM, use the DSNTWFG exec in job step DSNTIST. See the comment block for job step DSNTIST for information about the parameters for DSNTWFG.

If you do not run the installation CLIST, customize DSNTIJTM or the DSNTWFG exec to create at least one 4 KB and one 32 KB work file table space.

Related concepts:
- [Work file database storage requirements](#)

---

**Installation step 19: Define and bind Db2 objects: DSNTIJSG**

Use job DSNTIJSG to bind Db2-supplied packages and plans, initialize the Db2 communications database, and create user-managed databases for the Db2 resource limit facility (RLF), Data Definition Control support, and query optimization.

**Before you begin**

**Important**: In a data sharing environment, you must ensure that the resource limit facility (RLF) is inactive on all members in the data sharing group before running DSNTIJSG. To do this, issue the STOP RLIMIT command for each member.

**Procedure**

To use job DSNTIJSG:

1. If you use a product that uses a semicolon as a delimiter, replace the semicolons at the end of each SQL statement in job DSNTIJSG before you run the job.

   If you use a product that uses a semicolon as a delimiter, the CLIST adds SQL statements to job DSNTIJSG. Products that use a semicolon as a delimiting character cause semicolons to be removed from the installation CLIST before it is executed.

2. Run job DSNTIJSG.

   If the DSNTIJSG job runs successfully, it produces return codes of 0. It might also produce a return code of 4 because a step within this job attempts to delete a row from a table that might not exist at the time that this job runs. Expect the following messages from the BIND statement for each object that Db2 provides:

   - DSNE932I WARNING, ONLY IBM-SUPPLIED PLAN NAMES SHOULD BEGIN WITH DSN
   - DSNE932I WARNING, ONLY IBM-SUPPLIED PACKAGE-IDS SHOULD BEGIN WITH DSN
   - DSNE932I WARNING, ONLY IBM-SUPPLIED COLLECTION-IDS SHOULD BEGIN WITH DSN

3. If the DSNTIJSG job fails or abends:
   a. Be sure that the user that is specified on the JOB statement is authorized. Use the same name that you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP1.
   b. Correct any other problems with the DSNTIJSG job, and run it again. If you do not have enough resources to run the job, review the values that you specified for the Db2 installation parameters. (See job DSNTIJUZ.) Use the
standard update procedure to make any necessary modifications. Then stop Db2, run the DSNTIJUZ job, start Db2, and run the DSNTIJSG job again.

What to do next

During the ISPF tailoring session, you named one or two IDs to have installation SYSADM authority. One of these users can now grant various levels of authority to other users. You can use SPUFI or a job similar to DSNTIJSG to perform the authorization. To grant SYSOPR authority to all primary and secondary user IDs that issue Db2 commands from the z/OS console or TSO SDSF, issue the following statement in SPUFI:

GRANT SYSOPR TO userid

Related concepts:

- Data definition control support database (Introduction to Db2 for z/OS)
- The resource limit facility (Introduction to Db2 for z/OS)

Related tasks:

- Populating the communications database for use with TCP/IP
- Special packages and plans for SPUFI
- Updating subsystem parameter and application default values
- Verifying successful Db2 12 installation or migration with sample applications

Related reference:

- DSNTIP5: Distributed data facility panel 2

Job DSNTIJSG

During installation, job DSNTIJSG binds Db2-supplied packages and plans.

When you run job DSNTIJSG during installation, it binds Db2 plans, including the plans for SPUFI and DCLGEN. DSNTIJSG also binds a package for a message routine that is used by SPUFI and DCLGEN, and binds that package into the plans for SPUFI and DCLGEN. If you use SPUFI to access remote sites, you might need to bind different packages and plans for SPUFI. You also might need to bind different packages and plans for SPUFI if some users need to run SPUFI under different terminal CCSIDs.

Job DSNTIJSG also binds the package and plan for the DSNTXAZP program, which updates a specified CLIST defaults input member (DSNTIDxx) with the current subsystem parameter and buffer pool settings that are defined in the subsystem.

The DSNTIJSG job also completes the following tasks for user-maintained database activity:

- Creates the resource limit facility database.
- Within the resource limit facility database, creates a new resource limit facility table, DSNRLMTnm, and a new resource limit facility index, DSNMRLrn. By default, the step that creates this new table and index is commented out in the JCL. To have Db2 create DSNRLMTnm and DSNMRLrn, uncomment the step.
- Creates the data definition control support database.
- Creates the program authorization database, table space, table, and index.

The SYSIBM.DSNPROGAUTH program authorization table is used to verify that a program is authorized to use a plan. A sample INSERT statement is provided, which you can uncomment, customize, and execute to populate the table.
The DSNTIJSG job inserts a blank row into the communication database (CDB) table SYSIBM.LUNAME. The CDB holds tables that contain information about your connection with remote Db2 subsystems, and a blank row allows all SNA clients to access DDF. TCP/IP remote clients cannot be controlled by using the CDB. If you run DSNTIJSG again, SQLCODE -803 is issued after an INSERT request because the blank row already exists.

The TCP/IP ALREADY VERIFIED field of panel DSNTIP5 defines the minimum security requirements for all TCP/IP clients because inbound security requirements cannot be established on individual clients.

**Installation step 20: Populate the user-maintained databases (optional)**

Job DSNTIJSG creates user-maintained databases that you might need to populate.

**Procedure**

Populate the user-maintained databases, including the resource limit specification table, and the data definition control support tables. Although Db2 automatically creates the user-maintained databases, you must complete this step if you plan to use data definition control or the resource limit facility (RLF).

**Installation step 21: Configure Db2 for running stored procedures and user-defined functions**

Before you can use user-defined functions or the following types of stored procedures, you need to perform some set-up tasks:

- External stored procedures
- External SQL procedures
- Native SQL procedures that satisfy at least one of the following conditions:
  - The native SQL procedure calls at least one external stored procedure, external SQL procedure, or user-defined function.
  - The native SQL procedure is defined with ALLOW DEBUG MODE or DISALLOW DEBUG MODE.
- Db2-supplied stored procedures

All routines, except native SQL procedures, must run in an environment that is established by the z/OS Workload Manager (WLM). A set of Db2-supplied routines are installed by job DSNTIJRT and validated by job DSNTIJRV as part of the installation process. You can also define your own routines or use routines that are provided by a non-IBM vendor. Before you can use any stored procedures and user-defined functions, however, you need to set up WLM environments for these routines. If you plan to use certain Db2-supplied stored procedures or user-defined functions, you need to perform some additional steps.

**Related concepts:**

- Stored procedures (Db2 Application programming and SQL)
- User-defined functions (Db2 SQL)

**Related tasks:**

- Installing the IBM Data Server Driver for JDBC and SQLJ on Db2 for z/OS (Db2 Application Programming for Java)
Tailoring Db2 subsystem parameters for stored procedures during installation (optional)

Several subsystem parameters should be tailored to your Db2 environment before you use stored procedures. These parameters were set when you installed or migrated Db2. If you are satisfied with those values, you do not need to complete this step.

About this task

The subsystem parameters for stored procedures are:

- MAX_NUM_CUR
- MAX_ST_PROC
- STORMXAB
- STORTIME

Procedure

To modify the values of these subsystem parameters, complete one of the following tasks:

- Run the installation CLIST again in INSTALL or MIGRATE mode. On the installation panels, leave most of the existing values and change the following fields on their respective panels:
  1. On panel DSNTIPA1, specify the input member that contains field values for your current installation.
  2. On panel DSNTIPT, specify different names for the TEMP CLIST LIBRARY and SAMPLE LIBRARY fields to avoid overwriting your original libraries.
  3. On panel DSNTIPX, add or change the values of the MAX ABEND COUNT, TIMEOUT VALUE, MAX OPEN CURSORS, and MAX STORED PROCS fields as appropriate.
  4. On panel DSNTIPY, specify a name for the REMOTE LOCATION field. Db2 uses this name for the stored procedure sample applications.

- Edit job DSNTIJUZ to add or change values for the subsystem parameters for routines.
  This method has the following disadvantages:
  - Db2 does not use these new values for any future migrations because your DSNTIDxx member and DSNTIJUZ parameters are not saved for future input.
  - Db2 does not generate the sample jobs for stored procedure and user-defined functions because you are not running the DSNTINST CLIST.

Related reference:

- DSNTIPA1 or DSNTILA1: Main panel
- DSNTIPT: Db2 data set names panel
- DSNTIPX: Routine parameters panel
- DSNTIPY: Job editing panel
- Directory of subsystem parameters, panel fields, and application default values

Setting up Language Environment for stored procedures during installation

*Language Environment* (LE) is a component of the z/OS operating system that establishes a common runtime environment for all applications, regardless of
programming language. Db2 uses Language Environment to provide a runtime environment for external stored procedures and external SQL procedures.

**About this task**

Language Environment provides a set of essential run time services that are available through a set of interfaces that are consistent across programming languages. These services include routines for run time message handling, condition handling, and storage management.

Language Environment provides the following advantages for Db2 stored procedures:

- Language Environment hides the differences between programming languages.
- Language Environment provides the ability to make a stored procedure resident in the stored procedure address spaces.
- Language Environment supports a large number of run time options, including those options that are needed to use tools to debug your stored procedures.

**Procedure**

To set up Language Environment for stored procedures:

2. Specify the Language Environment run time library in the JCL startup procedure for each WLM-managed address space for stored procedures. This library must be the only available run time library. Do not reference any other language run time libraries within the system link list or within the STEPLIB statements. If other language run time libraries are defined in the system link list, use STEPLIB statement overrides to exclude them.

Multiple languages can share the same Language Environment run time library. Therefore, you do not need to specify the language-specific libraries in the JCL startup procedure of each stored procedure address space.

**Restriction:** Do not use compilers (or associated run time libraries) that are no longer supported by IBM.

**What to do next**

After you set up Language Environment for stored procedures, you can override the Language Environment run time options for individual stored procedures. You can do this by using the RUN OPTIONS clause in the CREATE PROCEDURE statement or ALTER PROCEDURE statement. The default values for Language Environment run time options are defined by z/OS. You might have overridden some of these default values when you customized Language Environment.

**Related concepts:**


**Related reference:**

Implementing RRS for stored procedures during installation

To use some Db2 stored procedures, you must implement Resource Recovery Services (RRS). RRS is a sync point manager program that coordinates two-phase commit processing. Db2 requires that RRS be active because WLM-managed stored procedure address spaces use the RRS attachment facility (RRSAF).

About this task

RRS records events for protected resources in log streams that are managed by a z/OS component called the system logger. The following table identifies the log streams that RRS uses, where gname can be your Sysplex name or any name in a non-Sysplex environment:

Table 45. Log streams that RRS uses

<table>
<thead>
<tr>
<th>Log stream type</th>
<th>Log stream name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main unit-of-recovery log state stream</td>
<td>ATR.gname.MAIN.UR</td>
<td>The state of active units of recover (URs). RRS periodically moves this information into the RRS delayed UR state log when UR completion is delayed.</td>
</tr>
<tr>
<td>Delayed unit-of-recovery log state stream</td>
<td>ATR.gname.DELAYED.UR</td>
<td>The state of active URs, when UR completion is delayed.</td>
</tr>
<tr>
<td>Resource manager data log stream</td>
<td>ATR.gname.RM.DATA</td>
<td>Information about the resource managers that use RRS services.</td>
</tr>
<tr>
<td>Restart log stream</td>
<td>ATR.gname.RESTART</td>
<td>Information, which is needed during restart, about incomplete URs. This information enables a functioning RRS instance to assume remaining incomplete work from an RRS instance that failed.</td>
</tr>
<tr>
<td>Archive log stream</td>
<td>ATR.gname.ARCHIVE</td>
<td>Information about completed URs. This log stream is optional, but recommended.</td>
</tr>
</tbody>
</table>

All of these log streams are required except for the archive log stream. RRS does not start if it is not able to connect to the required log streams.

In a Sysplex environment, these log streams are shared by the systems of the Sysplex.

Procedure

To implement RRS for stored procedures:
1. Define the RRS log streams that are to be written, either on disk or in the coupling facility by completing the following actions:
   a. If you are using the coupling facility, add definitions for the RRS structures in the CFRM policy.
      When you are use coupling facility log streams, each log stream needs to be mapped to a coupling facility structure.

   **Example:** The following example shows JCL that updates the CFRM policy to add the RRS structures:
   ```jcl
   //DEFCFRM1 JOB MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
   //STEP1 EXEC PGM=IXCMIAPU
   //SYSPRINT DD SYSOUT=* 
   //SYSABEND DD SYSOUT=* 
   //SYSIN DD *
   
   DATA TYPE(CFRM) REPORT(YES)
   DEFINE POLICY NAME(CFRM18) REPLACE(YES)
   CF NAME(CF01)
   TYPE(009672)
   MFG(IBM)
   PLANT(02)
   SEQUENCE(000000040104)
   PARTITION(1)
   CPCID(00)
   DUMPSPACE(2048)
   CF NAME(CF02)
   TYPE(009672)
   MFG(IBM)
   PLANT(02)
   
   ................
   ................
   ................
   ................
   STRUCTURE NAME(RRS_ARCHIVE_1)
   INITSIZE(8000)
   SIZE(16000)
   PREFLIST(CF1,CF2)
   REBUILDPERCENT(5)
   STRUCTURE NAME(RRS_RMDATA_1)
   INITSIZE(8000)
   SIZE(16000)
   PREFLIST(CF1,CF2)
   REBUILDPERCENT(5)
   STRUCTURE NAME(RRS_MAINUR_1)
   INITSIZE(8000)
   SIZE(16000)
   PREFLIST(CF1,CF2)
   REBUILDPERCENT(5)
   STRUCTURE NAME(RRS_DELAYEDUR_1)
   INITSIZE(8000)
   SIZE(16000)
   PREFLIST(CF1,CF2)
   REBUILDPERCENT(5)
   STRUCTURE NAME(RRS_RESTART_1)
   INITSIZE(8000)
   SIZE(16000)
   PREFLIST(CF1,CF2)
   REBUILDPERCENT(5)
   
   b. Define the log streams in the system logger policy by using the IXCMIAPU utility. This utility program is provided in the SYS1.MIGLIB system library.
      If you are using the coupling facility, also define the structures in the system logger policy. You can map each log stream to a single structure or you can map log streams of like data types to the same structure.
Example of defining coupling facility log streams: The following example JCL shows definitions for coupling facility structures and RRS log streams. In this case, each RRS log stream is mapped to a structure.

```
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DATA TYPE(LOGR) REPORT(YES)

DEFINE STRUCTURE NAME(RRS_ARCHIVE_1) LOGSNUM(1)
MAXBUFSIZE(64000) AVGBUFSIZE(262)

DEFINE STRUCTURE NAME(RRS_RMDATA_1) LOGSNUM(1)
MAXBUFSIZE(1024) AVGBUFSIZE(252)

DEFINE STRUCTURE NAME(RRS_MAINUR_1) LOGSNUM(1)
MAXBUFSIZE(64000) AVGBUFSIZE(158)

DEFINE STRUCTURE NAME(RRS_DELAYEDUR_1) LOGSNUM(1)
MAXBUFSIZE(64000) AVGBUFSIZE(158)

DEFINE STRUCTURE NAME(RRS_RESTART_1) LOGSNUM(1)
MAXBUFSIZE(64000) AVGBUFSIZE(158)

DEFINE LOGSTREAM NAME(ATR.SANDBOX.ARCHIVE) STRUCTNAME(RRS_ARCHIVE_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM NAME(ATR.SANDBOX.RM.DATA) STRUCTNAME(RRS_RMDATA_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM NAME(ATR.SANDBOX.MAIN.UR) STRUCTNAME(RRS_MAINUR_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM NAME(ATR.SANDBOX.DELAYED.UR) STRUCTNAME(RRS_DELAYEDUR_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM NAME(ATR.SANDBOX.RESTART) STRUCTNAME(RRS_RESTART_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)
/*

Example of defining disk log streams: The following example JCL shows definitions for RRS log streams on disk.

```
/DEFSZEXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DATA TYPE(LOGR)
```
DEFINE LOGSTREAM NAME(ATTRPLEX1.MAIN.UR)
  LOWOFFLOAD(60)
  HIGHOFFLOAD(80)
  DASDONLY(YES)
  HLQ(TEST)
  LS_SIZE(1024)
  LS_DATACLAS(VSAMLS)
  STG_SIZE(1024)

DEFINE LOGSTREAM NAME(ATTRPLEX1.DELAYED.UR)
  LOWOFFLOAD(60)
  HIGHOFFLOAD(80)
  DASDONLY(YES)
  HLQ(TEST)
  LS_SIZE(960)
  LS_DATACLAS(VSAMLS)
  STG_SIZE(960)

DEFINE LOGSTREAM NAME(ATTRPLEX1.ARCHIVE)
  LOWOFFLOAD(0)
  HIGHOFFLOAD(80)
  DASDONLY(YES)
  HLQ(TEST)
  LS_SIZE(960)
  LS_DATACLAS(VSAMLS)
  AUTODELETE(YES)
  RETPD(2)
  STG_SIZE(2000)

DEFINE LOGSTREAM NAME(ATTRPLEX1.RM.DATA)
  LOWOFFLOAD(60)
  HIGHOFFLOAD(80)
  DASDONLY(YES)
  HLQ(TEST)
  LS_SIZE(192)
  LS_DATACLAS(VSAMLS)
  STG_SIZE(192)

DEFINE LOGSTREAM NAME(ATTRPLEX1.RESTART)
  LOWOFFLOAD(60)
  HIGHOFFLOAD(80)
  DASDONLY(YES)
  HLQ(TEST)
  LS_SIZE(960)
  LS_DATACLAS(VSAMLS)
  STG_SIZE(960)

c. If you are using coupling facility log streams, compile and link-edit the changed CFRM policy.

d. If you are using coupling facility log streams, activate the updated CFRM policy in your Sysplex by using the following operator command:

   SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname

2. Establish the priority for RSS within the WLM policy.
   The RSS priority needs to be equal to or higher than the dispatching priority of Db2. You can use the SYSSTC service class for the RSS address space to achieve a higher dispatching priority.

3. Move the ATTRRS procedure that is supplied by IBM from SYS1.SAMPLIB to your SYS1.PROCLIB as member RRS, and make any necessary changes. This JCL procedure is used to start RRS.
   You can use a member name other than RRS, but the first four characters of this name must match the subsystem name that you specify in the IEFSSNxx member of SYS1.PARMLIB in the next step.
The GNAME value in the procedure must match the gname that you specified when defining the log streams. The default value for gname at RRS startup is the Sysplex name.

Example: The following example shows a sample procedure to start RRS.

```assembler
//RRS PROC GNAME='',CTMEM='
//RRS EXEC PGM=ATRIMIKE,REGION=0M,TIME=NOLIMIT,
//      PARM='GNAME=&GNAME
//CTMEM=&CTMEM'
//
```

4. Define RRS as a subsystem to z/OS by editing the active IEFSSN.xx member of SYS1.PARMLIB to include the following entry:

```
SUBSYS SUBNAME(RRS) /* RESOURCE RECOVERY SERVICES */
```

The subsystem name can be RRS or any other name that you choose. However, the first four characters (or fewer characters if the name is shorter) must match the name of the JCL procedure to start RRS.

5. Start RRS by using the following operator command:

```
START RRS
```

If you subsequently need to stop RRS, use one of the following operator commands:

- SETRRS CANCEL
- SETRRS SHUTDOWN

If neither of these commands stops RRS, use the FORCE RRS,ARM command. In this command, RRS is the subsystem name that your installation assigned to RRS in PARMLIB member IEFSSNxx.

Related reference:
- z/OS MVS Programming: Resource Recovery
- Configuring and defining RRS logging requirements (z/OS MVS Programming: Resource Recovery)

Related information:

Setting up a WLM application environment for stored procedures during installation

All stored procedures and user-defined functions (except native SQL procedures) must run in WLM-managed stored procedure address spaces. You must assign each stored procedure to a WLM application environment, which routes the stored procedure work to the appropriate WLM-managed address space.

About this task

If you use data sharing, see “Special considerations for WLM-managed stored procedures in coexistence” on page 704 for additional guidance about setting up WLM environments.

Recommendation: Use partitioned data set extended (PDSE) for load libraries that contain external modules for stored procedures. Using PDSEs might eliminate your
need to stop and start the stored procedures address space due to growth of the load libraries. If a load library grows from additions or replacements, the library might need to be extended.

**Procedure**

To set up a WLM application environment for one or more stored procedures:

1. Gather requirements for the WLM environment. These requirements include:
   - The name of the Db2 subsystem or data sharing group where the stored procedure or user-defined function is to run.
   - The maximum number of task control blocks (TCBs) that can run concurrently in an address space that is initiated by the WLM environment. This setting, referred to as the NUMTCB, is the maximum number of procedures or functions that should run concurrently in that WLM-managed address space. The appropriate setting depends on the requirements or limitations of the individual stored procedure or user-defined function.
     - For recommended NUMTCB values for Db2-supplied stored procedures, see [“Installation information for procedures and functions that are supplied with Db2” on page 829](#).
     - For recommended NUMTCB values for your own stored procedures, see [Assigning stored procedures and functions to WLM application environments (Db2 Performance)](#). If a routine allocates a large amount of memory or if the external module for the routine is not re-entrant, the NUMTCB value should be 1. If there is a need for serialization, the NUMTCB value should also be 1.
     - The WLM environment for a REXX stored procedure must have a NUMTCB setting of 1.
   - Requirements for Authorized Program Facility (APF).
     - Stored procedures and user-defined functions that execute privileged instructions (for example, WLM_REFRESH) must run in a WLM environment that has an APF-authorized address space.
     - Db2 REXX stored procedures, such as DSNTPSMP, and other routines that use TSO/E and REXX facilities must run in a WLM environment that does not have an APF-authorized address space.
     - Most stored procedures can run in a WLM environment that either has or does not have an APF-authorized address space. However, running a routine in an APF-authorized environment is recommended only when doing so is necessary.
   - Special data sets and DD names that are needed by the stored procedure or user-defined function. For example, all Java stored procedures implicitly require a Language Environment run time options data set to be allocated to DD name JAVAENV. Other routines might also have an explicit requirement for a work file to be allocated to a particular DD name.

2. Create a JCL startup procedure for the WLM application environment.

   Use `ssnmWLM` (where `ssnm` is the Db2 subsystem ID) as the basis for creating the new startup procedure. `ssnmWLM` is the default name of the sample JCL procedure for starting WLM-established address spaces. If you specified a name in the WLM PROC NAME field on the DSNTIPX installation panel, the sample procedure has that specified name. The sample procedure is placed in your system PROCLIB by installation job DSNTIJMV. For the DSNACICS stored procedure, DSNTIJMV also provides sample procedure `ssnmCICS`, where `ssnm` is the subsystem ID. Tailor your new startup procedure as follows:
• Change the APPLENV value to the name of the WLM application environment that you plan to define.

• Change the startup procedure name to an appropriate procedure name.

  **Recommendation:** Specify the same name for the application environment name and the procedure name. Using the same name simplifies monitoring the WLM environment definitions.

• Change the value of DB2SSN to your Db2 subsystem name.

• Ensure that the NUMTCB value is appropriate. If the minimum number of address spaces is specified, the NUMTCB value will be the number of task control blocks that are started in an address space.

  **Recommendation:** If you want to be able to easily change the NUMTCB value, specify the NUMTCB value here, and do not specify this parameter on the application environment definition panel when you define the WLM application environment. Specifying NUMTCB on the environment definition panel associates this parameter with the WLM application environment definition. Any changes to the JCL startup procedure can be made by refreshing the WLM application environment. However, changes to the WLM application environment require that you reinstall the z/OS service policy at an LPAR or sysplex level. If you specify the NUMTCB parameter for both the WLM application environment and the startup procedure, the value in the WLM application environment overrides the value in the startup procedure.

• If the environment is one in which the cost of starting new address spaces is very high, you can use the MNSPAS parameter. The MNSPAS parameter specifies the minimum number of address spaces that are to be started and maintained. The value can be 0 through 50. The default value is 0, which means that no minimum number of address spaces is specified. If the number of address spaces is at the specified minimum, cancelling an address space causes WLM to start another one to maintain the minimum. If you refresh a WLM environment, WLM starts a number of new address spaces as needed (or the specified minimum number) and stops the old address spaces.

  If the MNSPAS parameter is specified, the NUMTCB value is the number of task control blocks that are started in an address space. If the MNSPAS parameter is not specified, the NUMTCB value is the maximum number of task control blocks that can be started in an address space.

• Edit the data set names to match your data set names.

• Ensure that the library in which the stored procedure resides is in the STEPLIB concatenation.

• Ensure that the IBM Language Environment run time library is specified in the STEPLIB concatenation, unless your system includes the Language Environment run time library in the system link list.

• If you want to run the stored procedure in a WLM environment that has an APF-authorized address space, ensure that all the libraries in the STEPLIB concatenation are defined to the Authorized Program Facility.

• Add any special DD statements that are required for the stored procedures or user-defined functions that are to run in the WLM environment.
  – If you plan to use stored procedures that use the ODBA interface to access IMS databases, make the following additional changes:
- Add the data set name of the IMS data set that contains the ODBA callable interface code to the end of the STEPLIB concatenation. The name of this data set is usually IMS.RESLIB.

- After the STEPLIB DD statement, add a DFSRESLB DD statement that names the IMS data set that contains the ODBA callable interface code.

- If your routines are defined as PROGRAM TYPE SUB and use UNIX System Services (USS) syscall statements, you might need to add a NOUNDUB DD statement. By default, the USS process and its resources are cleaned up after every stored procedure and user-defined function invocation. If you need the task in this WLM environment to retain the process across subsequent invocations, add a dummy NOUNDUB DD statement. This option specifies that no cleanup will occur unless the WLM environment is refreshed. To enable this option, add the following statement to the WLM stored procedure address space startup JCL:

//NOUNDUB DD DUMMY

Use this option with caution because routines use the USS process that was used by the previous routine that was run under the same task. Also note that any open resources such as file descriptors remain open for the next routine that is run under the same task. Consider using this option on new workloads only if reusing the USS process is the desired behavior.

Example of a startup procedure: The following example shows a startup procedure for a WLM application environment for external SQL, COBOL, C, C++, PL/I, and assembler stored procedures. This startup procedure for external SQL stored procedures needs one unauthorized data set included in STEPLIB.

```java
//*******************************************************************
// JCL FOR RUNNING THE WLM-ESTABLISHED STORED PROCEDURES
//*******************************************************************
// ADDRESS SPACE
// RGN -- THE MVS REGION SIZE FOR THE ADDRESS SPACE.
// DB2SSN -- THE DB2 SUBSYSTEM NAME.
// NUMTCB -- THE NUMBER OF TCBS USED TO PROCESS
// END USER REQUESTS.
// APPLENV -- THE MVS WLM APPLICATION ENVIRONMENT
// SUPPORTED BY THIS JCL PROCEDURE.
//*******************************************************************
//DB9AWLM PROC RGN=0K,APPLENV=WLMENV,DB2SSN=DB9A,NUMTCB=4G,MNSPAS=3
//IEFPROC EXEC PGM=DSNX9WLM,REGION=RGN,TIME=NOLIMIT,
// PARM=’DB2SSN,NUMTCB,APPLENV,MNSPAS’
//STEPLIB DD DISP=SHR,DSN=DB9AU.RUNLIB.LOAD
// DD DISP=SHR,DSN=CEE.SCEERUN
// DD DISP=SHR,DSN=DB9A9.SDSNEXIT
// DD DISP=SHR,DSN=DB9A9.SDSNLOAD
```

3. Define the application environment to WLM by using one of the following methods.

- Use the WLM administrative application (IWMAIN0) to define a new environment or copy or modify an existing one.

  a. In the first WLM panel, Definition Menu, enter 9 for Application Environments.

  The application environment definition panel is displayed.

  b. On the application environment definition panel, specify the appropriate action code:

  - Create a new WLM application environment.
  - Copy an existing WLM application environment.
  - Modify an existing WLM application environment.
The appropriate panel for creating, copying or modifying the application environment is displayed.

c. If you are creating or modifying a WLM environment, specify the appropriate options for the WLM application environment definition, using the same application environment name and procedure name that you specified for the startup procedure.

d. After you specify the appropriate options, press Enter to create the application environment definition.

- Modify and run sample job DSNTIJRW. As provided, DSNTIJRW creates the core Db2 WLM environments, but you can modify the job to define your own WLM environments. Job DSNTIJMV copies an address space procedure for each of the Db2 core WLM environments into SYS1.PROCLIB.

4. Install the WLM service definition that contains information about the application environment by opening the WLM Definition Menu panel and selecting the Install function in the Utilities menu bar.

5. Activate the WLM service policy from the installed service definition by completing one of the following actions:

- Issue the VARY WLM command as follows:

  \texttt{VARY WLM,POLICY=\textit{policy-name}}

- On the WLM Definition Menu panel, choose the Activate service policy option from Utilities menu bar.

6. Set up security for the WLM application environments and for creating stored procedures by completing the following actions:

   a. Optional: Specify which WLM-established address spaces can run stored procedures. If you do not complete this step, any WLM-established address space can run stored procedures.

   b. Grant access to users to create procedures in specific WLM address spaces.

   c. Grant access to users to create procedures in specific schemas. Use the GRANT statement with the CREATIN option for the appropriate schema.

   d. Grant access to users to create packages for procedures in specific collections. Use the GRANT statement with the CREATE option for the appropriate collection.

   \textbf{Requirement}: For native SQL procedures, the collection name for the package must be the same as the schema name.

   e. Grant access to refresh the WLM environments to the appropriate people.

   For more detail, see \textcolor{blue}{Managing authorization for stored procedures (Managing Security)}

7. Ensure that the ID that is used to run the JCL startup procedure for the WLM application environment has permission to use RRSAF. Each time one of the Db2 WLM address spaces is started, it uses RRSAF to attach to Db2.

\textbf{What to do next}

When you create or alter a stored procedure, specify the WLM application environment that Db2 is to use in the WLM\_ENVIRONMENT option of the CREATE or ALTER PROCEDURE statement. When you specify environments, be aware that:

- Routines that have low memory requirements and that are frequently called should not share an environment with routines that have high memory requirements and are rarely called. Otherwise, storage is wasted.
• Stored procedures can be grouped according to the authorization settings for a WLM environment.
• WLM environments should not be shared among routines that use the same DD name for different purposes.
• WLM environments should not be shared among routines if one routine allocates data sets in the environment and that routine is run only occasionally. If other routines that do not use the data sets use the environment, the data sets are kept artificially allocated, which might interfere with other tasks.

Related concepts:
- WLM-established stored procedures address spaces storage requirements

Related tasks:
- Managing authorization for stored procedures (Managing Security)
- Refreshing WLM application environments for stored procedures (Db2 Administration Guide)

Related reference:
- Core WLM environments for Db2-supplied routines
- DSNTWLMB tool
- GRANT (collection privileges) (Db2 SQL)
- GRANT (schema privileges) (Db2 SQL)
- set_dub_default (BPX1SDD, BPX4SDD) (z/OS UNIX System Services Programming): Assembler Callable Services Reference

Related information:
- Using Operator Commands for Application Environments (z/OS MVS Planning: Workload Management)

Installation step 22: Set up Db2-supplied routines

Installation jobs are provided for the setup of all Db2-supplied routines and supporting objects.

Setting up the Db2 core WLM environments during installation

Db2 stored procedures (except native SQL procedures) must run in WLM-managed stored procedure address spaces. You must assign each stored procedure to a WLM application environment, which routes the stored procedure work to the appropriate WLM-managed address space.

About this task

For Db2-supplied routines, you can use the provided definitions for core WLM environments as a starting point for defining your environments. You can also manually define the WLM environments by using the IWMARIN0 application, as described in Setting up a WLM application environment for stored procedures during installation.

Exception: You do not need to complete this configuration task for native SQL procedures unless they call at least one external stored procedure or external SQL procedure. This task also applies to native SQL procedures that call at least one user-defined function or that are defined with ALLOW DEBUG MODE or...
DISALLOW DEBUG MODE. Otherwise, this task applies to only user-defined functions, external stored procedures, external SQL procedures, and Db2-supplied stored procedures

**Procedure**

To set up the provided core WLM environments:

Run job DSNTIJRW.
Job DSNTIJRW invokes the DSNTWLMB tool to create the core WLM environments. The CLIST assigns the core WLM environment names in DSNTIJRW according to the values that are specified on panel DSNTIPR1. For detailed information about the environments that are created by DSNTIJRW, see “Core WLM environments for Db2-supplied routines” on page 836.

**Related concepts:**
WLM management of stored procedures

**Related reference:**
DSNTWLMB tool
DSNTIPR1: Parameters for Db2-supplied routines panel

### Installing Db2-supplied routines during installation

The DSNTIJRT and DSNTIJRV jobs install and validate the installation of Db2-supplied routines. These jobs are configured with the options that you specified on installation panel DSNTIPR1 and panels DSNTIPRA through DSNTIPRP.

**Before you begin**

If the RACF facility class is active and a profile was defined for MVSADMIN.WLM.POLICY, the user that runs the DSNTIJRV job requires READ access. To grant READ access to a user, an administrator must submit the following command:

```
PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY) ID(user) ACCESS(READ)
```

If the RACF OPERCMDS class is active and a profile was defined for MVS.MCSOPER.*, the user ID that runs the DSNTIJRV job requires READ access. Alternatively, you can add a discrete profile for the DSNTRVFY console that is used in this job. To add a discrete profile for the DSNTRVFY console, an administrator must complete the following steps:

1. Establish a discrete profile for MVS.MCSOPER.DSNTRVFY by issuing the following command:
   ```
   RDEFINE OPERCMDS MVS.MCSOPER.DSNTRVFY UACC(NONE)
   ```
2. Grant READ access to the user that runs the DSNTIJRV job by issuing the following command:
   ```
   PERMIT MVS.MCSOPER.DSNTRVFY CLASS(OPERCMDS) ID(user) ACCESS(READ)
   ```
3. Refresh the OPERCMDS class by issuing the following command:
   ```
   SETROPTS RACLST(OPERCMDS)REFRESH
   ```

Resource Recovery Services (RRS) is also required for these jobs.
Procedure

To install the Db2-supplied routines:

1. Run job DSNTIJRT to install and configure all Db2-supplied routines.
2. After DSNTIJRT successfully completes, run job DSNTIJRV to validate the installation of the routines.
3. Authorize the appropriate users to use one or more of these stored procedures by issuing the GRANT EXECUTE statement.

   As part of the installation process, job DSNTIJRT defines, binds, and grants access to the stored procedures and user-defined functions. Job DSNTIJRT grants EXECUTE authority to the authorization IDs that are specified for the GRANT EXECUTE fields on installation panels DSNTIPRB through DSNTIPRP. The default setting for GRANT EXECUTE is PUBLIC, meaning that anyone who can access Db2 can use these stored procedures.

   **Recommendation:** When you run the installation CLIST, use the GRANT EXECUTE fields on panels DSNTIPRB through DSNTIPRP to specify different access lists.

4. Complete additional setup steps for any of the following routines that you plan to use.
   - The stored procedures for XML schema support
   - The administrative task scheduler routines and administrative enablement routines
   - The CICS transaction invocation procedure
   - The common SQL API stored procedures
   - The WebSphere® MQ user-defined functions
   - The web service user-defined functions

What to do next

After installing the Db2-supplied routines, you can choose to disable any of the routines that are unnecessary or unusable on your subsystem.

Related concepts:
- Configuration control statements for Db2-supplied routines

Related tasks:
- Granting privileges for executing stored procedures and stored procedure packages (Managing Security)
- Disabling Db2-supplied routines for validation

Related reference:
- Core WLM environments for Db2-supplied routines

Job DSNTIJRT

Job DSNTIJRT installs and configures Db2-supplied routines by executing program DSNTRIN.

DSNTIJRT provides a configuration control statement for each Db2-supplied routine and the mode for processing those statements. The configuration control statement specifies the WLM environment, list of authorization IDs that are allowed to execute the routine, and optional package owner for the routine. DSNTRIN reads and validates those statements and uses them to create and customize routines and supporting objects.
Format

DSNTRIN accepts the following parameters:

**DB2SSN**

Identifies the Db2 subsystem on which to install and configure the Db2-supplied routines.

**MODE**

During SQL processing, DSNTRIN configures each routine with the WLM environment specified in the configuration control statement for that routine. Processing behavior depends on the DSNTRIN run mode, the Db2 catalog mode, and which objects already exist.

**INSTALL**

This is the normal operating mode. Use it when installing or migrating to Db2.

Create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the routine already exists, it is updated, if necessary, to use the WLM environment specified in the configuration control statement for that routine. If the package for a routine already exists, it is not bound again unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or created global temporary table. If a supporting object is used by more than one routine, the object is created when one of those routines is first created.

**REINSTALL**

This mode behaves like INSTALL mode, except that existing Db2-supplied routines and created global temporary tables are dropped ahead of other processing. Existing related databases, table spaces, tables, and indexes are not dropped, except as discussed in the special notices section of the DSNTIJRT job prolog.

Use REINSTALL mode only if you have a specific need to drop and re-create all Db2-supplied routines. Before running the job, verify that each GRANTTO statement in DSNTIJRT specifies all users who are authorized to use that routine.

In particular, use this mode to create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the package for a routine already exists, it is not rebound unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or created global temporary table. Existing routines and any related created global temporary tables are first dropped before they are re-created. Related databases are not dropped if they already exist, but they are created if they do not exist. If a supporting object is used by more than one routine, the object is created when one of those routines is first created. Drop processing of routines and related objects that do not exist is bypassed.

**REINITIALIZE**

This mode behaves like REINSTALL mode, except that existing databases, table spaces, tables, and indexes used by the Db2-supplied routines are also dropped ahead of other processing.
REINITIALIZE mode drops and reinitializes databases such as the XSR database that may contain information needed by Db2 for other processing. Therefore, use REINITIALIZE mode only when advised to do so by IBM Support. Before running the job, verify that each GRANTTO statement in DSNTIJRT specifies all users who are authorized to use that routine.

In particular, use this mode to create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the package for a routine already exists, it is not rebound unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or created global temporary table. Existing routines, existing related created global temporary tables, and existing related databases are first dropped before they are re-created. After each SQL statement, a COMMIT statement is issued. If a supporting object is used by more than one routine, the object is created when one of those routines is first created.

**INSTALL-PREVIEW**
Simulate INSTALL mode without updating the system. Generated JCL with embedded DDL and bind statements is outputted to the JCLOUT DD statement.

**REINSTALL-PREVIEW**
Simulate REINSTALL mode without updating the system. Generated JCL with embedded DDL and bind statements is outputted to the JCLOUT DD statement.

**REINITIALIZE-PREVIEW**
Simulate REINITIALIZE mode without updating the system. Generated JCL with embedded DDL and bind statements is outputted to the JCLOUT DD statement.

**AUTHID**
DSNTRIN uses this value as the CURRENT SQLID for creating and configuring most Db2-supplied routines. If DEFPKOWN is not specified, DSNTRIN also uses this value as the default OWNER for binding packages for the Db2-supplied routines that do not have the PKGOWNER specified in the configuration control statement.

**SECDEFFID**
DSNTRIN uses this value as the CURRENT SQLID setting when creating and configuring Db2-supplied routines that are defined with the SECURITY DEFINER option.

The ID that you specify for SECDEFFID must be an ID that can be used as a logon ID, because WLM uses the ID to execute Db2-supplied routines that are defined with SECURITY DEFINER.

**DEFPKOWN**
Specifies the default OWNER for binding the package for a Db2-supplied routine when the PKOWNER parameter of the routine's configuration control statement is not specified. If DEFPKOWN is not specified, DSNTRIN uses the AUTHID value as the default package owner.

The DEFPKOWN parameter is required only when the AUTHID parameter is set to SYSINSLTL. If the INSTALL PKG OWNER field on panel DSNTIPG is specified, the installation CLIST uses that value to set DEFPKOWN during the editing of DSNTIJRT. Otherwise, it uses the value specified in the ROUTINES
CREATOR field on panel DSNTIPG. If SYSINSTL is specified on panel DSNTIPG, the installation CLIST adds a notice to the DSNTIJRT prolog to explain additional requirements.

Specify an installation system administrator ID if you use the installation SYSOPR authority to execute this job for Db2 installation. Specify an authorization ID that is granted the system DBADM and DATAACCESS authorities if you use the installation SYSOPR authority to execute this job for Db2 migration.

**Input**

DSNTRIN allocates the following DD statements for input:

**DBRMLIB**

Location of the DBRMs for Db2-supplied routines. This is typically the prefix.SDSNDBRM target library.

**SYSUT1**

Internal work file for the Db2 installation bind interface (DSNTRBI).

**CFGIN**

Location of the configuration control statements for Db2-supplied routines.

**DB2OPT**

This required DD can contain the following optional parameters:

**STOGROUP**

Specifies a storage group to use for creating databases and table spaces that are used by the routines that are supplied with Db2. If this parameter is not specified, the default SYSDEFLT storage group is used.

**INDEXSTOG**

Specifies a storage group to use for creating indexes instead of the default SYSDEFLT storage group.

**BP4K**

Specifies an alternate 4-KB buffer pool for base table spaces that are created by DSNTRIN. The default 4-KB buffer pool is BP0.

**BP8K**

Specifies an alternate 8-KB buffer pool for base table spaces that are created by DSNTRIN. The default 8-KB buffer pool is BP8K0.

**BP16K**

Specifies an alternate 16-KB buffer pool for base table spaces that are created by DSNTRIN. The default 16-KB buffer pool is BP16K0.

**BP32K**

Specifies an alternate 32-KB buffer pool for base table spaces that are created by DSNTRIN. The default 32-KB buffer pool is BP32K.

**LOBBP8K**

Specifies an alternate 8-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 8-KB buffer pool is BP8K0.

**LOBBP16K**

Specifies an alternate 16-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 16-KB buffer pool is BP16K0.

**LOBBP32K**

Specifies an alternate 32-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 32-KB buffer pool is BP32K0.
IMS_SECURITY
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied IMS routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 IMS routines are altered if needed to conform with the SECURITY type that is specified here.

MQ_SECURITY
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied MQ routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 MQ routines are altered if needed to conform with the SECURITY type that is specified here.

SOAP_SECURITY
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied web services/SOAP routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 SOAP routines are altered if needed to conform with the SECURITY type that is specified here.

Note: DSNTRIN does not create or activate this storage group or any of these buffer pools.

Output
DSNTRIN allocates the following DD statements for output:

SYSPRINT
Destination for echo of configuration control statements that are read and for messages that are generated during processing. Each processed routine generates at least one status message.

SYSTSPRT
Internal message file for the Db2 installation bind interface (DSNTRBI).

CFIGOUT
Destination for configuration control statements that are read by DSNTRIN. This output is useful for diagnostic purposes when DSNTRIN reports an error in a configuration control statement.

SQLOUT
Destination for DDL statements that are generated and processed by DSNTRIN. These statements are useful for diagnostic purposes.

BINDOUT
Destination for bind statements that are generated and processed by DSNTRIN. These statements are useful for diagnostic purposes.

JCLOUT
Destination for a generated sample job that is generated when one of the PREVIEW modes is specified. The job contains steps that process inline DDL and bind statements that are needed to install and configure missing, downlevel, or incorrectly configured Db2-supplied routines. This DD is required if you specify a PREVIEW mode.

Example of executing DSNTRIN in INSTALL mode:

```plaintext
//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
   //   PARM=('DB2SSN(DSN) MODE(INSTALL)',
   //       'AUTHID(ADMF001) SECDEFID(ADMF002)',
   //       'DEFPKOWN(SYSADM)'),

//DBRMLIB DD DISP=SHR,DSN=DSN!!0.SDSNDBRM
//SYSUT1 DD UNIT=SYSDA,SPACE=(27930,(10,5)),
```

Chapter 5. Installing Db2 463
Installation and Migration Guide

Example of executing DSNTRIN in INSTALL-PREVIEW mode:

```
//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
//    PARM=('DB2SSN(DSN) MODE(INSTALL-PREVIEW)',
//    'AUTHID(ADMF001) SECEDEFID(ADMF002)',
//    'DEFPKOWN(SYSADM)')

//DBRMLIB DD DISP=SHR,DSN=DSN!!0.SDSNDBRM
//SYSUT1 DD UNIT=SYSDA,SPACE=(27930,(10,5)),
//    DCB=(RECFM=FB,LRECL=133)
//SYSPRINT DD SYSOUT**,DCB=(RECFM=FB,LRECL=133)
//SYSTSPT DD SYSOUT**
//CFIGOUT DD SYSOUT**
//SQLOUT DD SYSOUT**
//BINDOUT DD SYSOUT**
//JCLOUT DD DSN=USER.JCLLIB(DB2INST),DISP=SHR
//DB2OPT DD *
STOGROUP(DB2_ROUTINES_GROUP)
INDEXSTOG(DB2_INDEX_GROUP)
BP4K(BP0)
BP8K(BP8K0)
BP16K(BP16K0)
BP32K(BP32K)
LOBBP8K(BP8K0)
LOBBP16K(BP16K0)
LOBBP32K(BP32K0)
IMS_SECURITY(DB2)
MQ_SECURITY(DB2)
SOAP_SECURITY(DB2)
```

```
GRANTTO(PUBLIC)

SYSROC.DSNWZP
WLMENV(DSNWLM_NUMTCB1)
GRANTTO(PUBLIC)

/*

Related concepts:
Configuration control statements for Db2-supplied routines
Installation and migration jobs that the CLIST tailors

Related reference:
Core WLM environments for Db2-supplied routines

Job DSNTIJRV
Job DSNTIJRV validates the Db2-supplied routines by executing program DSNTVRFY. You should validate these routines after Db2 installation and after each phase of migration.

DSNTVRFY validates a single routine, multiple routines, or all Db2-supplied routines. DSNTVRFY produces a report that identifies routines that passed the validation and possible problem areas for routines that failed.

Authorizations

If the RACF facility class is active and a profile was defined for MVSADMIN.WLM.POLICY, the user that submits this job requires READ access.

If the RACF OPERCMDS class is active and a profile was defined for MVS.MCSOPER.*, the user ID that submits this job requires READ access.
Alternatively, you can add a discrete profile for the DSNTVRFY console that is used in this job.

Format

DSNTVRFY accepts the following required parameters:

**DB2SSN(ssid)**
Identifies which Db2 subsystem to connect to for validating the Db2-supplied routines.

**ROUTINE(name)**
Identifies which Db2-supplied routine needs to be validated. The following values are supported for name:
- A routine name in the form of schema-name.routine-specific-name.
- The keyword **DD:SYSIN** to specify a list of routines. List the routines in the form of schema-name.routine-specific-name under the SYSIN DD statement with one routine per line.
- The keyword **ALL** to specify all Db2-supplied routines.

DSNTVRFY also accepts the following optional parameters:

**AUTHID(authorization ID)**
DSNTVRFY uses this value as the CURRENT SQLID for issued SQL statements. Also, if PKOWNER(pkowner) is not specified, DSNTVRFY uses this value as the OWNER for BIND PACKAGE and BIND PLAN statements. If this value is not specified, the primary authorization ID is used for BIND PACKAGE, BIND PLAN, and issued SQL statements.
**PKOWNER(pkowner)**

Specifies the default owner for binding the DSNTRVFY plan and package. If the INSTALL PKG OWNER field on panel DSNTIPG is specified, the installation CLIST uses that value to set PKOWNER(pkowner) during the editing of DSNTIJRV. Otherwise, it uses the value specified in the ROUTINES CREATOR field on panel DSNTIPG. If SYSINSTL is specified on panel DSNTIPG, the installation CLIST adds a notice to the DSNTIJRV prolog to explain additional requirements.

If you use an authorization ID with the installation SYSOPR authority to execute job DSNTIJRV, the ID must be explicitly granted the MONITOR2 system privilege. If you set AUTHID to SYSINSTL for running the SYSPROC.WLM_REFRESH stored procedure, you must also define SYSINSTL in RACF.

**Output**

DSNTRVFY allocates the following required DD statements for output:

**SYSPRINT**
Contains messages generated during verification processing. Each processed routine can generate one or more messages.

**SYSTSPRT**
DSNTRVFY writes bind output to SYSTSPRT if an error occurs during bind processing.

The following JCL example is for validating a single routine:

```jcl
//DSNTJVFY EXEC PGM=DSNTRVFY, 
// PARM=’DB2SSN(VA1A) ROUTINE(SYSPROC.DSNUTILS)’ 
//DBRMLIB DD DSN=DB2A.SDSNDBRMRM,DISP=SHR 
//SYSUT1 DD UNIT=SYSDA,SPACE=(32000,(30,30)),DCB=(RECFM=VB,LRECL=133) 
//SYSTSPRT DD SYSOUT** 
//SYSPRINT DD SYSOUT** 
//SYSIN DD * 
```

The following JCL example is for validating multiple routines:

```jcl
//DSNTJVFY EXEC PGM=DSNTRVFY, 
// PARM=’DB2SSN(VA1A) ROUTINE(DD:SYSIN)’ 
//DBRMLIB DD DSN=DB2A.SDSNDBRMRM,DISP=SHR 
//SYSUT1 DD UNIT=SYSDA,SPACE=(32000,(30,30)),DCB=(RECFM=VB,LRECL=133) 
//SYSTSPRT DD SYSOUT** 
//SYSPRINT DD SYSOUT** 
//SYSIN DD * 
SYSPROC.DSNWZP 
SYSPROC.DSNACCOR 
SYSPROC.DSNUTILS 
SYSPROC.DSNUTILU 
```

The following JCL example is for validating all routines:

```jcl
//DSNTRVFY EXEC PGM=DSNTRVFY,COND=(4,LT), 
// PARM=('DB2SSN(DSN) AUTHID(SYSINSTL) ', 
// 'PKOWNER(ADMF001) ROUTINE(ALL)') 
//DBRMLIB DD DSN=DB2A.SDSNDBRMRM,DISP=SHR 
//SYSUT1 DD UNIT=SYSDA,SPACE=(32000,(30,30)),DCB=(RECFM=VB,LRECL=133) 
//SYSTSPRT DD SYSOUT** 
//SYSPRINT DD SYSOUT** 
//SYSIN DD * 
```
Processing

DSNTRVFY completes the following validation:

- Verifies that the routine is defined to the SYSIBM.SYSROUTINES catalog.
- Verifies that the collection ID defined in the SYSIBM.SYSROUTINES catalog table matches the collection ID used to bind the package.
- Verifies that the routine is currently defined with the Db2-reserved WLM application environment name DSNWLM_NOT_ENABLED, indicating that it is not enabled to run on the current Db2 subsystem. If it is, then a warning message is issued and the validation is bypassed.
- Verifies if the routine can be executed on the current Db2 catalog mode.
- Verifies the APF authorization requirements.
- Verifies if the WLM application environment is defined and available for use.
- Verifies if the NUMTCB setting is valid or within the recommended range.
- Verifies if the required DD statements for some of the routines are defined.
- Verifies if the WLM environment assigned to Java routines is enabled for Java.
- Verifies security coexistence requirements (for example, some routines should run on only their own WLM application environment).
- Verifies that the selected routine can be called successfully. Certain routines such as DSNACICS, DSNAIMS, MQSeries functions, and most debugger routines are restricted to basic validation and are not called.

During the running of this job, the following message might be issued on the system console:

IEFC452I DB2UDSMD - JOB NOT RUN - JCL ERROR

In most cases, this message is normal and can be disregarded. If you use IBM Optim™ Development Studio, see [Debugging stored procedures on Db2 for z/OS with Optim Development Studio, Part 2 (IBM developerWorks)](https://www.ibm.com) for information about DB2UDSMD.

If the validation is successful, the following message is issued:

DSNTO27I DSNTRVFY VALIDATION FOR schema.routine-specific-name SUCCESSFUL

If the validation fails, messages are issued to explain the reasons for failure. Examine these messages and make necessary adjustments before running the validation job again.

Input

DSNTRVFY requires the following DD statements for input:

**DBRMLIB**

Identifies the library where the DBRM for the DSNTRVFY program is located. This is typically the prefix.SDSNDBRM target library.

**SYSIN**

Contains the list of Db2 routines. This is required when the validation type is for multiple routines (DD:SYSIN).

**SYSUT1**

Work data set required for bind processing.
Limitations

The following limitations apply to the DSNTRVFY program:

- The name of the routine in the form `schema-name.routine-specific-name` is limited to 72 characters, the maximum line length in JCL.
- Only Db2 12 Db2-supplied routines are supported for validation.
- Some routines, if enabled, receive basic validation to ensure that they are defined correctly and assigned to a valid WLM application environment. However, validation will not extend to calling these routines for one or more of the following reasons:
  - Optional system support such as CICS, IMS, or MQSeries is required.
  - External setup on a remote server or workstation is required.
  - A call could introduce or modify data unexpectedly in a production environment.

The Db2-supplied routines that are restricted to basic validation are:

- `DB2DEBUG.CREATET_SESSION`
- `DB2DEBUG.DESTROY_SESSION`
- `DB2DEBUG.GET_REPORT`
- `DB2DEBUG.LIST_SESSION`
- `DB2DEBUG.PUT_COMMAND`
- `SYSFUN.SYSTS_ENCRYPT`
- `SYSFUN.XSLTRANSFORM`
- `SYSPROC.ADMIN_COMMAND_UNIX`
- `SYSPROC.ADMIN_JOB_CANCEL`
- `SYSPROC.ADMIN_JOB_FETCH`
- `SYSPROC.ADMIN_JOB_QUERY`
- `SYSPROC.ADMIN_JOB_SUBMIT`
- `SYSPROC.DBG_ENDSESSIONMANAGER`
- `SYSPROC.DBG_INITIALIZECLIENT`
- `SYSPROC.DBG_RECVCLIENTREPORTS`
- `SYSPROC.DBG_SENDCLIENTREQUESTS`
- `SYSPROC.DBG_SENDCLIENTCOMMANDS`
- `SYSPROC.DBG_TERMINATECLIENT`
- `SYSPROC.DSNACICS`
- `SYSPROC.DSNAIMS`
- `SYSPROC.DSNAIMS2`
- `SYSPROC.DSNLEUSR`
- `SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY`
- `SYSPROC.SYSTS_ALTER`
- `SYSPROC.SYSTS_CREATE`
- `SYSPROC.SYSTS_DROP`
- `SYSPROC.SYSTS_RESTORE`
- `SYSPROC.SYSTS_START`
- `SYSPROC.SYSTS_STOP`
- `SYSPROC.SYSTS_TAkeOVER`
- `SYSPROC.SYSTS_UPDATE`
– All MQSeries, web services, and SOAP functions (those with schema DB2MQ, DB2MQ1C, DB2MQ2C, or DB2XML)

Notes

Temporary stored procedures used by DSNTIJRV
DSNTRFY creates, calls, and drops two temporary stored procedures. These temporary stored procedures are SYSPROC.DSNWLM_INSTALL_INFOJ and SYSPROC.DSNWLM_INSTALL_INFO. Both of these stored procedures exist only when DSNTRFY is executing. DSNTRFY calls SYSPROC.DSNWLM_INSTALL_INFOJ to validate the WLM environment that is assigned to each Db2-supplied Java routine. DSNTRFY calls SYSPROC.DSNWLM_INSTALL_INFO to validate the WLM environment that is assigned to each Db2-supplied non-Java routine that runs in a WLM environment.

Using DSNTIJRV to verify 64-bit Java routines
Java on z/OS has two types of architectures: a 31-bit version and a 64-bit version. If your Java routines run in a 31-bit Java virtual machine (JVM), the startup procedure for the WLM environment in which the routines run must initialize a 31-bit environment. If you are running Java routines in a 64-bit JVM, the startup procedure for the WLM environment in which your Java routines run must initialize a 64-bit environment. The program that initializes a 31-bit WLM environment is DSNX9WLM. The program that initializes a 64-bit WLM environment is DSNX9WJM.

For example, you might use the Db2 core Java WLM environment, DSNWLM_JAVA, for your Java routines. By default, the DSNWLMJ startup procedure for DSNWLM_JAVA runs program DSNX9WLM. If your JAVA_HOME environment variable specifies the path to a 31-bit JVM, such as /usr/lpp/java/IBM/J8.0, you need to run program DSNX9WLM to initialize the WLM environment. However, if your JAVA_HOME variable specifies a path to a 64-bit JVM, such as /usr/lpp/java/8.0_64, you need to run program DSNX9WJM to initialize the WLM environment. If you switch to a 64-bit JVM for your Java stored procedures, but do not update the initialization program to DSNX9WJM, DSNTIJRV fails.

Related concepts:
[Display a link to the related concepts page]

Installation step 23: Back up the Db2 directory and catalog: DSNTIJIC

For backup purposes, create a copy of the Db2 directory and catalog.

About this task

Attention: If you do not create this copy, Db2 can start. However, if errors that require you to reinstall Db2 occur in the directory or catalog and you do not have backup copies, you lose all your tables and data.

Recommendation: Copy the catalog and directory at least daily if you make any changes in them. Recovery time for these databases is longer if the copies are not current, and the entire subsystem is affected.
Procedure

To create an image copy of the directory and catalog:

1. Examine job DSNTIJIC for the following information:
   - The device type or unit name. The default value is SYSDA. If this is incorrect for your site, correct it.
   - Expiration date or retention period. You can add a retention period or an expiration date to the job.
   - The user on the JOB statement. Ensure that the user is authorized. If Db2 is started under ACCESS(MAINT), this must be the same user that you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP1.

2. Run job DSNTIJIC.
   The DSNTIJIC job contains a list of all the Db2 directory and catalog table spaces. When you run job DSNTIJIC, it invokes the Db2 image copy utility to copy these table spaces to disk or tape, depending on the value that you select for **IC DATA SET UNIT NAME** field. Having copies of table spaces enables you to recover the Db2 catalog and Db2 directory in case of a failure.

3. If the DSNTIJIC job fails or abends, verify that no problems exist with the tape or disk setup for image copy. If you find no problems with the setup, examine the utility job output (JOBLOG) or the console log for problems. For example, look for I/O errors or incorrect sizes.

What to do next

Run the DSNTIJIC job periodically, perhaps daily or weekly, to reduce the amount of time required for recovering the directory or catalog. The copied data and log data sets are needed for recovery.

Related concepts:

- Privileges and authorities (Managing Security)

Related tasks:

- Installation step 19: Define and bind Db2 objects: DSNTIJSC
- Migrating an existing data sharing group to Db2 12

Installation step 24: Set up the administrative task scheduler

You can use the administrative task scheduler to execute administrative tasks according to a time-based or event-based schedule.

Before you begin

You should have run job DSNTIJRA and job DSNTIJIN before you set up the administrative task scheduler.

About this task

**Restriction:** After migration to Db2 12, the administrative task scheduler is disabled until you run job DSNTIJRT.

The following figure shows an overview of the steps for installing and configuring the administrative task scheduler.
To set up the administrative task scheduler:

1. Customize the administrative task scheduler address space startup procedure (xxxxADMT) in job DSNTIJMV. The name of the startup procedure needs to match the value of the ADMTPROC subsystem parameter. Before using the startup procedure, locate and review the settings for the following parameters:

**DB2SSID**

The name of this Db2 subsystem.

**DFLTUID**

The default ID that is used by the administrative task scheduler to execute its tasks. DFLTUID must be different than the ID that is used to start this address space. DFLTUID must be an ID that can be used as a logon ID and has a password. DFLTUID cannot be a group ID.

**TRACE**

Indicates whether to activate tracing for the administrative task scheduler. The value can be ON or the default value of OFF.

You can specify other parameters, such as ERRFREQ, MAXHIST, MAXTHD, or STOPONDB2STOP in the startup procedure similarly to the three parameters DB2SSID, DFLTUID and TRACE.
ERRFREQ
Specifies how frequently message DSNA679I displays on the console, indicating that one of the redundant, active copies of the task list is not accessible. By default, this message displays on the console once per minute, for example, when Db2 is offline.

MAXHIST
Specifies the number of status entries per task that are stored. This parameter is a positive integer with a default value of 10. When the limit is reached, the oldest status entries are deleted.

MAXTHD
Specifies the maximum number of execution threads for an administrative task scheduler. The default value is 99.

STOPONDB2STOP
Specifies that the administrative task scheduler will terminate when Db2 is stopped.

Also locate and review the ADMTDD1 DD statement. The VSAM data set that is indicated must match the VSAM task list data set that is defined in job DSNTIJIN.

The following example shows how these parameters can be updated.

```
//DSNADMT PROC LIB='DSN!!0.SDSNLOAD',
// DB2SSID=DSN,
// DFLTUID=DFLTUID,
// TRACE=OFF,
// MAXTHD=10
//*
//STARTADM EXEC PGM=DSNADMT0,DYNAMNBR=100,REGION=0K,
// PARM=('DB2SSID=&DB2SSID',
// ' DFLTUID=&DFLTUID',
// ' TRACE=&TRACE'
// ' MAXTHD=&MAXTHD'
// ' ERRFREQ=1440'
// ' STOPONDB2STOP')
//STEPLIB DD DISP=SHR,DSN=&LIB
//ADMTDD1 DD DISP=SHR,DSN=DSNC!!0.TASKLIST
```

2. Ensure that the administrative task scheduler routines were enabled. They were enabled by job DSNTIJRT.

3. Create a job to make image copies of the administrative task scheduler table spaces. Establish a schedule for making image copies. In general, you should back up important databases on a regular basis. The database for the administrative task scheduler is no exception and should be copied on the same frequency as the Db2 catalog and directory. The following example shows how to copy the table spaces in this database to stacked data sets on tape with a retention period of 99 days:

```
//DSNTIC EXEC PGM=DSNUTILB,PARM='DSN,IMAGCOPY',COND=(4,LT)
//SYSPRINT DD SYSOUT=
//SYSUDDUMP DD SYSOUT=
//DSNTRACE DD SYSOUT=
//SYSLISTD DD *
LISTDEF DSNLDEF
INCLUDE TABLESPACES DATABASE DSNADMDB
//SYSIN DD *
TEMPLATE DSNTPLT
DSN(prefix.IMAGCOPY.&DB..;TS.);
DISP (NEW,KEEP,DELETE)
UNIT TAPE
STACK YES
```
Installation step 25: Verify successful installation of Db2 12

You can run the sample applications to verify that your installation process was successful.

Procedure

To verify successful installation:

Tailor and run the jobs for the sample applications, as described in Verifying successful Db2 12 installation or migration with sample applications. You can select the phases that you need to run based on the attachment facilities you installed, the languages you use, and whether the sample objects exist.

What to do next

You might want to enable more capabilities in the newly installed Db2 subsystem. For instructions, see Working with additional capabilities for Db2.
Chapter 6. Migrating Db2 to Db2 12

Migration is the process of upgrading from Db2 11 to Db2 12. As part of the migration process, you also activate function level 500 or higher.

Before you begin

Complete the following tasks:

- Prepare your system
- Load the Db2 libraries
- Use the installation CLIST to tailor the Db2 12 jobs for your environment

About this task

The Db2 migration process uses a single catalog level.

Overview of Db2 12 new function availability

The availability of new function depends on the type of enhancement, the activated function level, and the application compatibility levels of applications. In the initial Db2 12 release, most new capabilities are enabled only after the activation of function level 500 or higher.

Virtual storage enhancements

Virtual storage enhancements become available at the activation of the function level that introduces them or higher. Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release. That is, activation of function level 500 introduces no virtual storage enhancements.

Subsystem parameters

New subsystem parameter settings are in effect only when the function level that introduced them or a higher function level is activated. Most subsystem parameter changes in the initial Db2 12 release take effect in function level 500. For a list of these changes, see Subsystem parameter changes in Db2 12 (Db2 for z/OS What's New?).

Optimization enhancements

Optimization enhancements become available after the activation of the function level that introduces them or higher, and full prepare of the SQL statements. When a full prepare occurs depends on the statement type:

- For static SQL statements, after bind or rebind of the package
- For non-stabilized dynamic SQL statements, immediately, unless the statement is in the dynamic statement cache
- For stabilized dynamic SQL statements, after invalidation, free, or changed application compatibility level

Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. That is, function level 500 introduces no optimization enhancements.

SQL capabilities

New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that
run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous Db2 releases, including Db2 11 and DB2 10. For details, see Application compatibility levels in Db2 (Db2 Application programming and SQL).

When you initially migrate to Db2 12, Db2 is at function level 100, and you cannot use new Db2 facilities. Use function level 100 to complete migration tasks and verify that the subsystem is stable. In a data sharing environment, use function level 100 to migrate all members of the group to Db2 12. When the subsystem or data sharing group is stable at function level 100, you can activate a higher function level. After any new function level is activated, fallback to Db2 11 is no longer possible.

What to do next

After migration to Db2 12 is complete, you can resolve application incompatibilities and rebind applications at the appropriate application compatibility so that they can use new SQL capabilities. You can do these steps when you are ready and satisfied that your environment is stable on Db2 12. For details see, Application compatibility levels in Db2 (Db2 Application programming and SQL).

Related tasks:
- Migration step 29: Activate function level 500
- “Automating Db2 migration by using z/OS Management Facility” on page 570

Related reference:
- What's new in Db2 12 function levels (Db2 for z/OS What's New?)

Related information:
- IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)

Migrating your Db2 subsystem to Db2 12

The migration process for Db2 12 has a single phase. However, most new capabilities and enhancements in Db2 12 remain unavailable until you issue an ACTIVATE command to activate function level 500 or higher.

Before you begin

Attention: If you do not follow the documented procedures, unpredictable results might occur after migration to Db2 12.

Ensure that your Db2 11 subsystem is running in new-function mode and is at the proper service level. For more information about identifying required service, see Required maintenance for Db2 12 installation or migration.

Important: Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied.
Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

About this task

When you start Db2, the code level of the starting Db2 is compared to the code level that is required by the current Db2 catalog. If the starting Db2 has a code-level mismatch with the catalog, Db2 does not start and a message is issued.

Refer to the [Db2 Program Directory](#) for keyword specifications that are used for Preventive Service Planning (PSP) and for information about the service levels of any coupling facilities. You can use the z/OS DISPLAY CF command to display the service level of coupling facilities. Check Information/Access or the ServiceLink facility of IBMLink for PSP information before you migrate. Or, to help you identify required service, you can instead use enhanced HOLDDATA and fix categories (FIXCATs), which provide a simplified, automated method of identifying and applying missing PTFs that are required for installation or migration. Also, check monthly for current information about Db2.

Restriction: After migration to Db2 12, the administrative task scheduler is disabled until you run job DSNTIJRT.

Procedure

To migrate a Db2 for z/OS subsystem to Db2 12:

Complete the following tasks:

- “Migration step 1: Actions to complete before migration” on page 478
- “Migration step 2: Verify the integrity of Db2 table spaces (optional)” on page 485
- “Migration step 3: Identify invalid packages (optional)” on page 487
- “Migration step 4: Check for consistency between catalog tables (optional)” on page 487
- “Migration step 5: Take image copies of the directory and catalog: DSNTIJIC” on page 488
- “Migration step 6: Connect Db2 to TSO” on page 489
- “Migration step 7: Connect Db2 to IMS (optional)” on page 492
- “Migration step 8: Connect CICS to Db2 (optional)” on page 492
- “Migration step 9: Stop Db2 11 activity” on page 493
- “Migration step 10: Back up your Db2 11 volumes (optional)” on page 495
- “Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM” on page 495
- “Migration step 12: Establish subsystem security (optional)” on page 497
- “Migration step 13: Define Db2 12 to IRLM and MVS: DSNTIJMV” on page 498
- “Migration step 14: update Db2 address space start and language procedure definitions: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ” on page 500
- “Migration step 15: Define system data sets: DSNTIJIN” on page 502
- “Migration step 16: Define user authorization exit routines: DSNTIJEX (optional)” on page 502
- “Migration step 17: Refresh the z/OS definition of Db2” on page 504
Migration step 1: Actions to complete before migration

Before you migrate to Db2 12, you must prepare your system for migration. Preparations include making adjustments for release compatibilities, running premigration queries, and completing other premigration tasks.

Run premigration queries (DSNTIJPM)

Before you migrate to Db2 12, run premigration queries on the Db2 11 catalog.

Procedure

To run premigration queries:

1. Customize job DSNTIJPM as described in the job prolog.
2. Run job DSNTIJPM.

Results

Job DSNTIJPM generates reports about conditions that require action before or during the migration process. Such conditions include obsolete object formats, missing required objects, packages that must be rebound before migration, and dependencies that are affected by the migration process. For detailed information about the contents of the reports for migration to Db2 12, see the report descriptions in job DSNTIJPM.
You can find the complete DSNTIJPM job in IBM Knowledge Center.

**Avoid automatic binds at migration to Db2 12**

By taking appropriate actions before migration to Db2 12, you can minimize the risk of disruptive automatic binds of application plans and packages during and after migration to Db2 12.

**Before you begin**

Identify plans and packages that are subject to automatic bind at migration to Db2 12 because they were last bound in a release earlier than DB2 10. To do so, run job DSNTIJPM and examine the premigration reports about package copies and plans that are not supported after migration to Db2 12. For more information, see Run premigration queries (DSNTIJPM).

**Important**: Apply APAR PI87675 to eliminate the possibility of repeating automatic remigration binds during release coexistence.

**About this task**

Migration-related automatic binds occur in Db2 12 because it cannot use runtime structures from a plan or package that was last bound in a release earlier than DB2 10. Plans and packages that were bound in Db2 11 can run in Db2 12, without the risk of migration-related automatic binds. However, plans and packages that are bound in Db2 12 cannot run on Db2 11 members without an automatic bind in Db2 11.

If automatic binds are allowed to occur, they are inevitably disruptive for rolling online migrations, or any time that different releases coexist in data sharing environments. During and after migration, such automatic binds can result in costly problems, including performance regression and even application outages.

Automatic binds occur the first time that any plan or package runs in Db2 12, which means that the automatic binds, and resulting problems and application outages, can occur well after a migration window. That is, such problems and resulting outages can occur any time in the “first impression” window in Db2 12, which is any time until the plan or package first runs in Db2 12.

The best approach is to bind all packages and plans that might be subject to migration-related automatic binds in Db2 11 with good performance before you start the migration to Db2 12. Doing so reduces the risk of the following problems that can result from automatic binds:

- Application failures because of contention between releases. For any migration that brings into release coexistence, automatic binds might fail for plans or packages that are in use by a Db2 11 member, causing applications to fail on the Db2 12 member.

  Successful automatic binds on Db2 12 in a coexistence environment also create a package or plan that Db2 11 members cannot run without another automatic bind. The automatic bind might also fail if the plan or package is in use by a Db2 12 member.

- Application failures, when automatic binds fail for access control authorization exit users because the authorization ID for the automatic bind has insufficient privileges, compared to the owner of the package or the plan.

- Difficult to resolve performance regressions. For packages, a migration-related automatic bind also destroys the current package. If the automatic rebind
succeeds but introduces an access path regression, REBIND SWITCH is not possible. Any performance regression that occurs in Db2 12 might also occur in Db2 11.

**Procedure**

To avoid automatic binds during or after the Db2 12 migration process, take the following actions:

1. In Db2 11 well before migration to Db2 12, rebind any plans that job DSNTIJPM identifies. Automatic binds of plans can be particularly disruptive for application workloads that depend on few or even a single large plan.

2. In Db2 11 well before migration to Db2 12, rebind any packages that job DSNTIJPM identifies, and specify the PLANMGMT(EXTENDED) bind option. Free old package copies regularly, and especially just before the migration, to get recent original package copies. The prior package is preserved for use by Db2 11. If a performance regression occurs, you can rebind the package with the SWITCH option to recover the previous access path.

3. Set the ABIND subsystem parameter appropriately for your environment:
   - In data sharing, set ABIND to COEXIST. If a package or plan is bound in Db2 12 and it runs later in Db2 11, it is automatically rebound. However, the Db2 11 plan or package can be used when it runs again in Db2 12, without another rebinding. However, the plan or package runs without any Db2 12 enhancements in this case.
   - In non-data sharing Db2 subsystems, set ABIND to YES. Coexistence across releases is not an issue in such cases. If you must fall back to Db2 11, explicitly rebind packages and plans that were bound in Db2 12. By doing so, you can avoid remigration automatic binds that occur for packages or plans that were bound in Db2 12 and are then automatically bound after fallback to Db2 11.

4. Bind most packages or plans in Db2 12 only after migration is complete for all members and coexistence with Db2 11 is no longer a concern.

**Related concepts:**
- Automatic binds in coexistence
- Automatic rebinds (Db2 Application programming and SQL)

**Related reference:**
- AUTO BIND field (ABIND subsystem parameter)

**Free inactive package copies**

When you use plan management policies to save package copies, original copies can become stale. By freeing inactive copies just before migration, you enable Db2 to save usable original package copies at the first rebind in Db2 12.

**Before you begin**

You can do this task only if you previously specified a plan management policy to specify that Db2 saves access path information for SQL statements, as described in Saving access path information for static SQL statements (Db2 Performance). Use the default option EXTENDED in most cases.

**About this task**

Original package copies in particular can become stale or invalid because Db2 never replaces the original copies after you first rebind with EXTENDED, unless
you free them. An original copy that is many years old probably cannot be used. Because the goal is to keep stable and usable access paths available, it is best to free invalid copies before migration, so that Db2 saves a new original copy in the first rebind after migration.

**Procedure**

To enable Db2 to save usable inactive package copies in the first rebind after migration, take the following action:

Just before migration, issue FREE PACKAGE commands and specify PLANMGMTSCOPE(INACTIVE). Db2 removes all original and previous package copies from the Db2 catalog. Then at the first rebind after migration to Db2 12, Db2 saves the last current package copies from Db2 11 as the new original package copies in Db2 12.

**Related tasks:**
- Saving and switching to previous access paths (Db2 Performance)

**Related reference:**
- FREE PACKAGE (DSN) (Db2 Commands)

**Make adjustments for incompatible changes in Db2 12**

Some changes in Db2 12 might affect your Db2 operations after migrating.

**Procedure**

To prepare for incompatible changes that the Db2 12 release introduces:
- Incompatible changes in Db2 12 (Db2 for z/OS What's New?)
- Incompatible changes summary for function levels 501 and higher (Db2 for z/OS What's New?)
- Function that Db2 12 no longer supports (Db2 for z/OS What's New?)
- Deprecated function in Db2 12 (Db2 for z/OS What's New?)

**Examine all new and changed values for DB2I panels**

During a migration, the DB2I default panels DSNEOP01 and DSNEOP02 are not initialized with the values that are specified during the installation CLIST process. The DB2I panel variables in the ISPF profile from the previous release are used on the current release. Any customized DSNEPROF members are migrated from your previous release to Db2 12.

**Procedure**

To ensure that your customized values are still valid:

Examine any new or changed DB2I default panel values.

**Ensure that sample objects from your previous release are available**

If you no longer have the IVP jobs from your previous release, you need to run the previous release's installation CLIST to regenerate them.

**Procedure**

To ensure that sample objects from your previous release are available:
If you dropped your previous release’s sample database (by running job DSNTEJ0), run the previous IVP jobs from the previous release through phase 3 before you start the migration.

If you do not have the jobs from the previous release available during migration, you will not have a Db2-supported IVP to verify a successful migration.

**Convert the BSDS, Db2 catalog, and directory to 10-byte RBA and LRSN format**

If the bootstrap data set (BSDS) is not already converted to use the 10-byte RBA and LRSN fields, you must convert it before you start Db2 in Db2 12. By default, the BSDS of a Db2 11 subsystem is initialized to use 6-byte RBA and LRSN fields. You must convert the catalog and directory to the 10-byte format before or soon after you convert the BSDS to the 10-byte format.

**Before you begin**

Before you can convert the BSDS, the Db2 subsystem or data sharing member must be started in Db2 11 new-function mode.

**Attention:** In Db2 subsystems that are not data sharing members, if Db2 is already at risk of reaching the 6-byte RBA limit, it is strongly recommended that you first convert all catalog and directory objects, then convert all user objects to the 10-byte RBA format, before you convert the BSDS.

**Procedure**

To convert the BSDS, Db2 catalog, and directory to 10-byte RBA and LRSN format, complete the following steps:

- Run job DSNTIJCV to convert the Db2 catalog and directory table spaces and indexes to the 10-byte RBA or LRSN format. If you run job DSNTIJCV in Db2 11, ensure that it allocates the Db2 11 SDSNLOAD module. Do not rearrange the processing sequence and the table space groupings. The COPY and REORG steps use LISTDEF filtering to exclude table spaces that are already in extended format.

  Job DSNTIJCV is customized when you run the Db2 installation CLIST for migration, as described in [Tailoring Db2 12 installation and migration jobs with the CLIST](#).

- Convert the BSDS to use the extended 10-byte RBA and LRSN fields, as described in [Convert BSDS records to the extended 10-byte format: DSNTIJCB](#).

**Related concepts:**

- The extended 10-byte RBA and LRSN in Db2 12 (Db2 for z/OS What's New?)
- Bootstrap data set (Introduction to Db2 for z/OS)

**Related reference:**

- [DSNJCNVT (Db2 Utilities)](#)

**Convert BSDS records to the extended 10-byte format: DSNTIJCB:**

You must convert the BSDS to a new format that supports 10-byte RBA and LRSN fields before you start Db2 in Db2 12. By default, the BSDS of a Db2 11 subsystem is initialized in the 6-byte, pre-Db2 11 format.
Before you begin

Before you can convert the BSDS, the Db2 subsystem or data sharing member must be started in Db2 11 new-function mode.

Attention: In Db2 subsystems that are not data sharing members, if Db2 is already at risk of reaching the 6-byte RBA limit, it is strongly recommended that you first convert all catalog and directory objects, then convert all user objects to the 10-byte RBA format, before you convert the BSDS.

Attention: After the BSDS is converted to the 10-byte format, Db2 stops issuing messages to warn you about the risk of reaching the 6-byte RBA or LRSN limits. The increased size of all log records also accelerates progress toward the 6-byte RBA logging limit.

In Db2 subsystems that are not data sharing members, always convert all Db2 catalog, directory, and user objects to use the extended 10-byte RBA format before you convert the BSDS, especially if Db2 is close to reaching the logging limit for the 6-byte RBA. Failure to convert page sets to the 10-byte RBA format before Db2 reaches the 6-byte logging limit results in failed updates with reason code 00C2026D. No updates are allowed for any object that is still in the 6-byte format.

In data sharing, if any Db2 member is approaching the logging limit for the 6-byte RBA but the LRSN is not approaching the limit of the 6-byte range, converting the BSDS of just that member sufficient to resolve the immediate problem and prevent outages. However, if the LRSN is also approaching the end of the 6-byte range, you must continue and convert page sets to use the 10-byte format before the limit is reached.

At migration to Db2 12, you cannot start Db2 12 until the BSDS is converted to use the 10-byte RBA and LRSN formats. You can convert the BSDS before or during the Db2 12 migration process.

Procedure

To convert the BSDS to use the extended 10-byte RBA and LRSN format, complete the following steps:

1. Stop Db2.

   Tip: If you complete the BSDS conversion as part of the Db2 12 migration process, you can avoid an extra stop and start of Db2. To do so, complete the BSDS conversion sometime between Migration step 9: Stop Db2 11 activity and Migration step 18: Start Db2 12.

2. Run job DSNTIJCB and make sure that it allocates the Db2 11 SDSNLOAD module. The DSNTIJCB job for each Db2 subsystem or data sharing member is located in the prefix.NEW.SDNSAMP sample library that is generated when you run the installation CLIST in MIGRATE mode. Db2 runs the DSNJCNVT conversion utility to convert the bootstrap data set records to support 10-byte RBA and LRSN fields. For more information, see DSNJCNVT (Db2 Utilities).

3. Start Db2.

What to do next

If you have not already done so, first convert all page sets for the Db2 catalog, directory, and , as described in Convert the BSDS, Db2 catalog, and directory to
10-byte RBA and LRSN format. Then convert all user objects to support the 10-byte RBA and LRSN, as described in Converting page sets to the 10-byte RBA or LRSN format (Db2 Administration Guide). After Db2 reaches the hard logging limit, you might not be able to convert the catalog and directory page sets.

You must continuously monitor the RBA and LRSN values until all catalog, directory, and user objects are converted to the 10-byte RBA or LRSN format. Failure to convert page sets before the 6-byte soft logging limit is reached results in failed updates with reason code 00C2026D, and any objects still in the 6-byte format become read-only. RBA or LRSN values greater than x'F00000000000' indicate that your system is at risk of reaching the 6-byte logging limit.

For more information the RBA and LRSN logging limits, see What to do before RBA or LRSN limits are reached (Db2 Administration Guide). For instructions for converting page sets to the 10-byte format, see Converting page sets to the 10-byte RBA or LRSN format (Db2 Administration Guide).

Related concepts:
- Bootstrap data set (Introduction to Db2 for z/OS)
- The extended 10-byte RBA and LRSN in Db2 12 (Db2 for z/OS What’s New?)

Related reference:
- Job DSNTIJCB

Job DSNTIJCB:

Job DSNTIJCB supports the DSNJCNVT stand-alone utility for converting the BSDS records to support 10-byte RBA and LRSN fields.

DSNTIJCB consists of the following job steps:

**DSNTDEF**
- Activates IDCAMS to define new BSDS data sets with CLUSTER, DATA, and INDEX components under the temporary names `prefix.BSDS01.NEW` and `prefix.BSDS02.NEW`.

**DSNTCNVT**
- Activates DSNJCNVT to read records from the existing BSDS, converts them to support the extended RBA and LRSN format, and writes them into the `prefix.BSDS01.NEW` and `prefix.BSDS02.NEW` data sets.

**DSNTRENO**
- Renames the existing BSDS data sets to backup names, `prefix.BSDS01.OLDFMT` and `prefix.BSDS02.OLDFMT`.

**DSNTRENN**
- Renames `prefix.BSDS01.NEW` and `prefix.BSDS02.NEW` to `prefix.BSDS01` and `prefix.BSDS02`.

No job step can run unless all preceding job steps complete with return code 0.

When the installation CLIST customizes DSNTIJCB, it substitutes the BSDS names that are specified in fields COPY 1 NAME and COPY 2 NAME on installation panel DSNTIPH for `prefix.BSDS01` and `prefix.BSDS02`. The fields allow each name to support names up to 30 bytes in length, so no special handling is needed to rename the existing data sets.

Related concepts:
The extended 10-byte RBA and LRSN in Db2 12 (Db2 for z/OS What's New?)

Related tasks:
Convert BSDS records to the extended 10-byte format: DSNTIJCB

Ensure that no utility jobs are running
In Db2 12, you can restart or terminate a utility only on the same release on which it was started.

About this task
Any outstanding utilities prior to Db2 12 cannot be restarted or terminated after you have migrated to Db2 12 conversion mode.

Immediately before you begin the migration process, ensure that you do not have outstanding utility jobs.

Procedure
To ensure that you do not have outstanding utility jobs:
Issue the DISPLAY UTILITY(*) command.

Related reference:
DISPLAY UTILITY (Db2) (Db2 Commands)

Migration step 2: Verify the integrity of Db2 table spaces (optional)
Use the DSN1COPY and CHECK INDEX utilities to verify the integrity of the Db2 directory and catalog table spaces on the subsystem that you are migrating.

About this task
Important: This migration step identifies data integrity issues that might interfere with subsequent migration tasks.

Tip: For best results, use this process to check the Db2 catalog and directory regularly, even outside of the migration process.

Procedure
To verify the integrity of your Db2 directory and catalog table spaces:
1. Run the DSN1COPY utility with the CHECK option on all catalog table spaces to ensure that the table space pages are physically correct and that the catalog table spaces are clustered.
   When you run this utility on segmented table spaces or LOB table spaces, you might receive message DSN1985I. You can ignore this message.
   The segmented table spaces in the directory and catalog are:
   • DSNDB01.SPT01
   • DSNDB01.SYSUTILX
   • DSNDB06.SYSSDDF
   • DSNDB06.SYSSSTATS
   • DSNDB06.SYSSTR
The LOB table spaces in the directory and catalog are:
- DSNDB01.SYSDBDXA
- DSNDB01.SYSSPUXA
- DSNDB01.SYSSPUXB
- DSNDB06.SYSJAUXA
- DSNDB06.SYSJAUXB
- DSNDB06.SYSPLUXA
- DSNDB06.SYSPLUXB
- DSNDB06.SYSTSATX
- DSNDB06.SYSTSCHD
- DSNDB06.SYSTSCHR
- DSNDB06.SYSTSCTD
- DSNDB06.SYSTSCTR
- DSNDB06.SYSTSIXR
- DSNDB06.SYSTSIXT
- DSNDB06.SYSTSOBX
- DSNDB06.SYSTSPDT
- DSNDB06.SYSTSHPH
- DSNDB06.SYSTSPKX
- DSNDB06.SYSTSPTX
- DSNDB06.SYSTSPVR
- DSNDB06.SYSTSQRA
- DSNDB06.SYSTSSNL
- DSNDB06.SYSTSSXL
- DSNDB06.SYSTSTRT
- DSNDB06.SYSTSVAAD
- DSNDB06.SYSTSVAIT
- DSNDB06.SYSTSVRT
- DSNDB06.SYSTSVVTT

2. Run the CHECK DATA and CHECK INDEX utilities.

3. You should run the following query on your Db2 11 catalog tables to ensure that you do not have a STOGROUP that is defined with both specific and non-specific volume IDs. If the query returns any rows, the identified STOGROUPs have both specific and non-specific volume IDs. Table spaces in databases that use these STOGROUPs cannot be image copied or recovered until ALTER STOGROUP is used to remove volumes so that the STOGROUP has either specific or non-specific volume IDs.

This query is commented out in the Db2 12 member DSNTESQ of prefix.SDSNSAMP.
SELECT * FROM SYSIBM.SYSVOLUMES V1
  WHERE VOLID != 'x'
  AND EXISTS (SELECT * FROM SYSIBM.SYSVOLUMES V2
              WHERE V1.SGNAME = V2.SGNAME AND V2.VOLID = 'x')

Related reference:
- CHECK INDEX (Db2 Utilities)
- DSN1COPY (Db2 Utilities)

Related information:
- DSN1985I (Db2 Messages)

Migration step 3: Identify invalid packages (optional)

Migrating to Db2 12 renders some packages invalid.

Procedure

To identify which packages are invalid after migration to Db2 12:

Run the following query on the Db2 11 subsystem that you are migrating.

```
SELECT DISTINCT SUBSTR(STRIP(BNAME) || ' ', 1, 18),
  STRIP(DCOLLID) || '.' || STRIP(DNAME) || '.' || HEX(DCONTOKEN)
FROM SYSIBM.SYSPACKDEP
WHERE (BTYPE = 'I'
  AND BQUALIFIER = 'SYSIBM'
  AND BNAME IN ('DSNOTX01', 'DSNATX02'))
OR (BTYPE = 'T'
  AND BQUALIFIER = 'SYSIBM'
  AND BNAME IN ('SYSCONTROLS', 'SYSENVIRONMENT',
                'SYSINDEXES', 'SYSKEYS',
                'SYSPACKAGE', 'SYSPACKCOPY',
                'SYSRoutines', 'SYSTABLES',
                'SYSTRIGGERS'))
ORDER BY 1, 2
```

If you are not in a data sharing coexistence environment, after migration, you can explicitly rebind these packages or let Db2 rebind them automatically. If you are in a data sharing coexistence environment, rebind these packages before you migrate any members to Db2 12.

Related concepts:
- Changes that invalidate packages (Db2 Application programming and SQL)

Migration step 4: Check for consistency between catalog tables (optional)

The DSNTESQ queries check the logical correctness of the Db2 catalog.
Before you begin

Before you run these queries, complete "Migration step 2: Verify the integrity of Db2 table spaces (optional)" on page 485 to run the DSN1COPY utility with the CHECK option and the CHECK DATA and CHECK INDEX utilities.

About this task

Tip: For best results, use this process to check the Db2 catalog regularly, even outside of the migration process.

Procedure

To check for consistency between catalog tables:

Run the queries that are not commented out in member DSNTESQ of the prefix.SDSNSAMP library. You can execute the SQL statements in DSNTESQ from SPUFI or from a dynamic SQL program like DSNTEP2.
You can run the queries on the actual catalog tables or on “mirror” copies of the catalog tables. If you run the queries on the copies, use the comment lines in member DSNTESQ for guidance. By running queries on copies of the catalog table, you reduce contention on the catalog.

Related tasks:

Migration step 2: Verify the integrity of Db2 table spaces (optional)

Migration step 5: Take image copies of the directory and catalog: DSNTIJIC

For backup purposes, create a copy of the catalog and directory of the system that you are migrating.

About this task

Attention: If you do not create this copy, Db2 can start. However, if errors in the catalog or directory require you to fall back to your previous version, you risk losing some of your tables and data.

Procedure

To create an image copy of the directory and catalog, perform the following steps with the Db2 11 copy of DSNTIJIC:

1. Examine job DSNTIJIC for the following items, and modify as needed:
   - The tape unit name. The job lists the tape unit name as TAPE. If this is incorrect for your site, correct it. The name TAPE is also used as the unit name for the default archive log data sets.
   - Expiration date or retention period. You can add a retention period or an expiration date to the job.
   - The user on the JOB statement. Ensure that the user is authorized. If Db2 is started under ACCESS(MAINT), this must be the same user that you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIP1.

2. Run the Db2 11 DSNTIJIC job.

   Job DSNTIJIC contains a list of all the Db2 directory and catalog table spaces. When you run DSNTIJIC, it invokes the Db2 image copy utility to copy these
table spaces to tape. The copied table spaces allow you to recover the Db2
catalog and directory in case of a failure.

3. If job DSNTIJIC fails or abends, look for problems with the tape that is set up
for image copy. If you do not find a problem, examine the log for problems.
For example, look for incorrect size or I/O errors.

**What to do next**

After migration, periodically run the Db2 12 job DSNTIJIC against the Db2 12
directory and catalog, perhaps daily or weekly. This action reduces the amount of
time that is required for recovering the Db2 directory or catalog. The copied data
and log data sets are needed for recovery.

If you are remigrating, you need to complete one of the following actions:

- Change the names of the data sets in which the new image copies are to reside.
  (Migration image copies use the current data set names.)
- Run the MODIFY utility to remove the migration image copies. If you select this
  option, ensure that you are familiar with the MODIFY utility.

Related reference:

[MODIFY RECOVERY (Db2 Utilities)]

**Migration step 6: Connect Db2 to TSO**

Access to TSO is required to support the interactive component of Db2 (DB2I).
Access to TSO also allows batch applications to access Db2 when those batch
programs are executed under the TSO terminal monitor program (TMP).

**Procedure**

To attach Db2 to TSO:

1. Make Db2 load modules available to TSO and batch users.
2. Make Db2 CLISTs available to TSO and batch users.
3. Make panels, messages, and load modules available to ISPF and TSO.

**What to do next**

Save your TSO logon procedures and JCL from Db2 11 in case you need to fall
back from Db2 12.

**Make Db2 load modules available to TSO and batch users**

To connect Db2 to TSO, you must first make Db2 load modules available to TSO
and batch users. If `prefix.SDSNEXIT` and `prefix.SDSNLOAD` are not included in
your LNKLISTxx, you can skip this step.

**Procedure**

To make Db2 load modules available to TSO and batch users:

If you have not included `prefix.SDSNEXIT` and `prefix.SDSNLOAD` in your
LNKLISTxx, add JOBLIB or STEPLIB statements to your logon procedures and JCL
to ensure that you access the Db2 12 load modules. Include `prefix.SDSNEXIT` before
`prefix.SDSNLOAD` in your JOBLIB or STEPLIB concatenations.
What to do next

You can attach to multiple releases of Db2 with your existing TSO or CAF logon procedures without changing the load libraries for your applications. After you migrate completely to the latest level of Db2, you must update those procedures and jobs to point to the latest level of Db2 load libraries.

Make Db2 CLISTs available to TSO and batch users: DSNTIJVC

Job DSNTIJVC merges the tailored CLISTs from prefix.NEW.SDSNTEMP with unchanged CLISTs from prefix.SDSNCLST, and it places all CLISTs in prefix.NEW.SDSNCLST.

About this task

Tailoring changes can modify these CLISTs: DSNEMC01, DSNH, DSNU, and DSNHC. The DSNTINST CLIST reads these CLISTs from prefix.SDSNCLST, edits them, and places them in prefix.NEW.SDSNTEMP. You can modify the default values.

The DSNEMC01 CLIST uses the values that are specified on installation panel DSNTIPF and stores the results in the ISPF profile member DSNEPROF. You can migrate any customized DSNEPROF members from Db2 11 to Db2 12. However, you need to examine any new or changed default panel values to ensure that your customized values are still valid.

DSNTIJVC also converts the Db2 CLISTs from a fixed-block record format to a variable-block format, with a record length of 84 and a block size of 3120. A CLIST that has been converted from fixed-block format to variable-block format cannot be used as input to the DSNTINST CLIST; use the unedited version of the SDSNCLST data set, as created by SMP/E.

Procedure

To make the CLISTs available to TSO and batch users:
1. If you use fixed-block CLIST libraries, modify the DSNTIJVC job as follows:
   • Change the SYSIN DD to DUMMY.
   • Change the allocation of prefix.SDSNCLST to match the data control block (DCB) attributes of your other CLIST libraries.
2. Run job DSNTIJVC.
3. Either concatenate prefix.NEW.SDSNCLST with your existing CLIST libraries or copy prefix.NEW.SDSNCLST into an existing CLIST library.

What to do next

If you need to rerun job DSNTIJVC, first delete data set prefix.NEW.SDSNCLST, which is created by this job.

When corrective service is applied to a CLIST, SMP/E changes only the prefix.SDSNCLST data set. You need to redo any record format changes and reapply any special tailoring that is required. You also need to move the CLIST to prefix.NEW.SDSNCLST. Corrective service (program temporary fixes) for these CLISTs is sent with ++HOLD statements, calling your attention to the possibility of additional work.

Related concepts: | 490 | Installation and Migration Guide |
Installation and migration jobs that the CLIST tailors

Make DB2I panels, messages, and load modules available to ISPF and TSO

During migration, to connect the Db2 Interactive facility (DB2I) to TSO, you must make panels, messages, and load modules available to ISPF and TSO.

About this task

DB2I uses the ISPF PROFILE and SHARED variable pools for most panel variable fields. As a result, you can easily re-enter a panel when panel variables have previously been specified. For the Db2 subcommands that permit lists of plan names, package names, DBRMs, and ENABLE and DISABLE statements, DB2I provides ISPF to contain all the user-specified variables for these subcommand keywords.

DB2I creates and maintains a set of ISPF tables in a user-defined TSO data set that is allocated to a data set with a ddname of DSNETBLS. The DB2I-generated tables in this library are DSNCONNS, DSNDBRMS, and DSNPLPKN. The following table shows the library table member names and their contents.

Table 46. The Db2 ISPF table library

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNCONNS</td>
<td>ENABLE or DISABLE connection type and connection name variables that are referenced by plan or package name</td>
</tr>
<tr>
<td>DSNDBRMS</td>
<td>Subcommand DBRM member and LIBRARY name variables that are referenced by plan name</td>
</tr>
<tr>
<td>DSNPLPKN</td>
<td>Package list variables that are referenced by package name</td>
</tr>
<tr>
<td>DSNPATHS</td>
<td>Schema name variables that are referenced by plan or package name and that are to be included in the PATH keyword</td>
</tr>
</tbody>
</table>

DB2I uses ISPF table services to maintain individual ISPF tables within the DSNETBLS data set. For performance reasons, ISPF keeps this table library in an open state once an individual table has been updated. Attempts to close this data set by using the TSO FREE command results in error message IKJ5686I.

Procedure

To make panels, messages, and load modules available to ISPF and TSO:

1. If you added the DB2I libraries as part of Tailoring Db2 12 installation and migration jobs with the CLIST, you can skip this step. Otherwise, concatenate the Db2 ISPF libraries with the ISPPLIB, ISPTLIB, ISPSSLIB, and ISPMLIB DD statements in your logon procedures. You must also concatenate the Db2 ISPF libraries with these DD statements in any of your CLISTs where the libraries might be allocated. These libraries are prefix.SDNSPFPP, prefix.SDNSPFPM, prefix.SDNSPFPS, and prefix.SDNSPFE or prefix.SDNSPF. If you use English DB2I panels, concatenate SDNSPFPE to ISPPLIB. If you use Kanji DB2I panels, concatenate SDNSPFPK instead of SDNSPFPE.

```plaintext
PROFILE WTP MSGID
ALLOCATE DDNAME(SYSPROC) DSN('prefix.SDSSCLST') SHR REUSE
ALLOCATE DDNAME(ISPPLIB) DSN('prefix.SDNSPFPM' + 'ISP.SISPMENU') SHR REUSE
ALLOCATE DDNAME(ISPSSLIB) DSN('prefix.SDNSPFPP' 'prefix.SDNSPFPE' + 'ISP.SISPSENU') SHR REUSE
ALLOCATE DDNAME(ISPMLIB) DSN('prefix.SDNSPFPS' + 'ISP.SISPLIB' 'ISP.SISPENU') SHR REUSE
```
2. Create the DSNETBLS data set.
   When allocating this data set, you should assign the following DCB attributes, where \( n \) is any integer:
   - DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(\( n \)*LRECL)

   The following example shows how to set up an ALLOCATE statement to create the data set:
   ```
   ALLOC DA(DSNSPFT) NEW SP(11) TR DIR(10) +
   DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(800)
   F(DSNETBLS) REUSE
   ```

   The following example shows how to allocate an existing data set to the data set with the DSNETBLS ddname:
   ```
   ALLOC DA(DSNSPFT) F(DSNETBLS) REUSE
   ```

3. If you want to run the ISPF/CAF sample application that is provided with Db2, ensure that the data set `prefix.RUNLIB.LOAD` is included in the logon procedures or in the ISPLLIB concatenation. If you chose IBMCOB for your `LANGUAGE DEFAULT` on panel DSNTIPF, you should also include the data set `prefix.CEE.SCEERUN`.

   **Related concepts:**
   - Dynamic SQL and the ISPF/CAF application

   **Related tasks:**
   - [Make Db2 load modules available to TSO and batch users](#)

   **Related reference:**
   - [z/OS ISPF Messages and Codes](#)

### Migration step 7: Connect Db2 to IMS (optional)

Connecting Db2 to IMS is an optional part of migration. Connecting Db2 to IMS requires coordination with your company’s group that supports IMS systems.

**Procedure**

To connect the IMS attachment facility:
1. Make Db2 load modules available to IMS
2. Define Db2 to IMS
3. Define new programs and transactions to IMS
4. Prepare IMS applications for Db2
5. Depending on your site, you might also need to:
   - Define Db2 plans for IMS applications
   - Generate a user language interface

**Related tasks:**
- Connecting the IMS attachment facility

### Migration step 8: Connect CICS to Db2 (optional)

To connect Db2 to CICS, you need to disconnect and connect the CICS region.

**About this task**

A macro is supplied with CICS to define the connection between CICS and Db2 by using a resource control table (RCT).
Ensure that you coordinate the attachment facility connection with your CICS support group.

**Procedure**

To connect the CICS attachment facility:
1. Recalculate space requirements for the CICS attachment facility.
2. Define your CICS attachment facility parameters using the RCT.
3. Update the CICS system tables.
4. Update the CICS initialization JCL.
5. Coordinate Db2 and CICS security if necessary.
6. Prepare new CICS applications for Db2 if necessary.

**Related tasks:**

- Defining the CICS Db2 connection

**Migration step 9: Stop Db2 11 activity**

Before making Db2 12 operational, ensure that all work is stopped on the Db2 11 subsystem.

**Before you begin**

If you do not stop work on the Db2 11 subsystem or data sharing member that you are migrating, fallback procedures might fail.

**Procedure**

To stop work on the system that you are migrating:
1. Issue the following command, where `-DSN1` is the subsystem command prefix that is defined for Db2. You must have system administrator (SYSADM) or system operator (SYSOPR) authority to issue this command.

   ```bash
   -DSN1 STOP DB2 MODE(QUIESCE)
   ```

   The QUIESCE keyword allows Db2 to complete the processing of programs that are currently executing. This might require some processing time.

2. Issue the following command to allow only the system administrators and system operators that were defined during installation to access Db2:

   ```bash
   -DSN1 START DB2 ACCESS(MAINT)
   ```

   If Db2 does not start properly, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. For example, if the bootstrap data set (BSDS) does
not match the subsystem parameter values, ensure that the correct jobs were
run for DSNTIJUZ. Ensure that you started Db2 with the correct subsystem
parameter option.

3. Follow these steps to ensure that all Db2 11 work is stopped:
   • Make sure no units of recovery remain. Issue the following command:

        GUPI
        -DSN1 DISPLAY THREAD(*) TYPE(*)
        GUPI

      Then use -DSN1 RECOVER INDOUBT for any indoubt threads.
      • All utilities must be restarted or terminated on the version on which they
        were started. If you do not use data sharing, ensure that all utilities are
        completed or terminated on Db2 11. To ensure that no utility work remains,
        issue the following command:

        GUPI
        -DSN1 DISPLAY UTILITY(*)
        GUPI

      After you have determined the utilities that are running, you can let them
      complete processing or you can terminate the utility. To stop all utilities,
      issue the following command:
      -DSN1 TERM UTILITY(*)
      • Ensure that no table spaces and index spaces in the Db2 directory
        (DSNDB01) or the Db2 catalog (DSNDB06) have write error ranges or
deferr ed restart states. To determine existing restrictions, issue the following commands:

        GUPI
        -DSN1 DISPLAY DATABASE(DSNDB01) SPACENAM(*) RESTRICT
        -DSN1 DISPLAY DATABASE(DSNDB06) SPACENAM(*) RESTRICT
        GUPI

      You must have system administrator or system operator authority (specified
on installation panel DSNTIPP1) to issue this command.
      Recover any table spaces and index spaces that have write error ranges or
deferr ed restart states by using the RECOVER utility.

4. Stop Db2 by issuing the following command:

        GUPI
        -DSN1 STOP DB2 MODE(QUIESCE)
        GUPI
A user with installation system administrator (installation SYSADM) or installation system operator (installation SYSOPR) authority must enter this command.

What to do next

If you have not already converted the BSDS to extended format, you must complete the conversion, as described in Convert the BSDS, Db2 catalog, and directory to 10-byte RBA and LRSN format before you begin Migration step 18: Start Db2 12⁴ on page 505.

Related reference:
- DISPLAY DATABASE (Db2) (Db2 Commands)
- DISPLAY THREAD (Db2) (Db2 Commands)
- DISPLAY UTILITY (Db2) (Db2 Commands)
- STOP DB2 (Db2) (Db2 Commands)
- TERM UTILITY (Db2) (Db2 Commands)

Migration step 10: Back up your Db2 11 volumes (optional)

After you have stopped Db2 11 activity, you can back up your Db2 11 subsystem. This backup provides a recovery base in case one becomes necessary.

Procedure

To back up your subsystem:

Take dumps of the Db2 subsystem data sets. You also can take dumps of the SMP/E data sets and the Db2 distribution and target libraries.

Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM

For migration to Db2 12, you must define initialization parameters in the subsystem parameter module (DSNZPxxx) and application defaults load module (dsnhdecp), and define the offline message generator CCSID (DSNHMCID).

Before you begin

If the Db2 subsystem that you are migrating has only one bootstrap data set (BSDS), add second BSDS before you migrate. For more information, see Adding a second BSDS.

About this task

The subsystem parameter module (DSNZPxxx) and application defaults module (dsnhdecp) names are specified in the following DSNITPO3 installation panel fields:

- PARAMETER MODULE field
- APPL DEFAULTS field

Procedure

To define the Db2 12 initialization parameters, complete the following steps:
1. Run job DSNTIJUZ. Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix:SDSNEXIT and prefix:SDSNLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This action ensures that members DSNSYSP, DSNJARVP, DSNJLOGP, DSNTPRPM, DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```
SET BOUNDARY (GLOBAL).
LIST DODEF (SMPTLIB).
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions.

For more information, see Job DSNTIJUZ: define the Db2 data-only subsystem parameter module.

2. Run job DSNTIJUA. Job DSNTIJUA defines the Db2 data-only application defaults module.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix:SDSNEXIT and prefix:SDSNLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUA to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUA so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZQ. This action ensures that member DSNARIB is linked with the he application defaults module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```
SET BOUNDARY (GLOBAL).
LIST DODEF (SMPTLIB).
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

For more information, see Job DSNTIJUA: define data-only application defaults module.

3. Run job DSNTIJUM. Job DSNTIJUM defines the Db2 data-only offline message generator CCSID (DSNHMCID) module which is needed for message conversion by Db2 applications and utilities. For more information, see Job DSNTIJUM: define Db2 data-only offline message generator CCSID (DSNHMCID) module.
Adding a second BSDS

If you are migrating a Db2 subsystem that has only one bootstrap data set (BSDS), add second BSDS to before you migrate.

**About this task**

A second BSDS simplifies recovery in most situations. With a second BSDS, you can continue working in many cases that otherwise require recovery and restart. The required storage is small, and the data set is relatively inactive.

**Important:** If you do not add a second BSDS, add TWOBSDS=NO in the DSN6LOGP macro.

**Procedure**

To add a second BSDS, complete the following steps:

1. Run job DSNTIJUZ to set the TWOBSDS=YES parameter in the DSN6LOGP macro. For more information, see [Job DSNTIJUZ: define the Db2 data-only subsystem parameter module](#).

   You might receive message GIM65001W when you run steps DSNTLOG and DSNTIMQ, or you might receive a return code of 4 when you run step DSNTIMQ. You can ignore these messages.

   If DSNTIJUZ fails or abends, correct the problem and rerun the job, using the same subsystem parameter name.

2. Define a second BSDS. Use the VDSNTISAM BSDS definition in job DSNTIJIN as an example. For more information, see [Job DSNTIJIN](#).

3. Add a //BSDS2 DD statement to the DSN1MSTR Db2 startup procedure.

4. Issue a RECOVER BSDS command to establish the dual BSDS. For more information, see [RECOVER BSDS (Db2) (Db2 Commands)](#).

**Related concepts:**

- Backup and recovery resources and tools (Introduction to Db2 for z/OS)

**Related tasks:**

- Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM

**Related information:**

- [Using Access Method Services (DFSMS Access Method Services for Catalogs)](#)

**Migration step 12: Establish subsystem security (optional)**

Db2 lets you control access to data within Db2. Db2 also works together with outside security systems, such as RACF, that control access to the Db2 subsystem.

**About this task**

Because your Db2 12 subsystem reuses the data objects from your Db2 11 subsystem, you have probably already supplied the protection that those objects
need. However, you probably want to protect the new (Db2 12) Db2 data objects.

Procedure

To establish subsystem security:

1. Customize job DSNTIJRA, as described in the job prolog.
2. Run job DSNTIJRA. Job DSNTIJRA defines the administrative task scheduler started task module to RACF program control and defines the administrative task scheduler as a trusted context in RACF. DSNTIJRA is intended for the site security administrator, and can be run any time prior to installing or migrating to Db2 12.
3. If you have a Db2 data sharing environment, customize and run DSNTIJRA for each member of the group.

Related concepts:

[Getting started with Db2 security (Managing Security)]

Migration step 13: Define Db2 12 to IRLM and MVS: DSNTIJMV

For migration to Db2 12, job DSNTIJMV updates SYS1.PARMLIB members to update the definition of Db2 12 to IRLM and MVS.

Before you begin

Ensure that you completed the task in Make Db2 load modules available to TSO and batch users.

About this task

Job DSNTIJMV demonstrates how to update SYS1.PARMLIB members to update the definitions for Db2 12 to IRLM and MVS. Although you might edit and run this job, the more common approach is to examine the changes that DSNTIJMV makes and use an editor, such as ISPF/PDF, to make the required updated directly to the SYS1.PARMLIB members.

Because DSNTIJMV cannot anticipate all the necessary updates to identify Db2 to MVS and IRLM in your z/OS environment, additional updates might be required for your site. You can complete these updates either by making the updates directly in SYS1.PARMLIB members, or by editing the DSNTIJMV job.

Do not overwrite the entries for Db2 and IRLM in the program properties table (PPT). Note that PPT entries should no longer be defined in SYS1.PARMLIB(SCHEDxx). Default entries are supplied in the SYS1.LINKLIB(IEFSDDPPT) load module, which is provided by z/OS. For details of the PPT entries, see Program properties table (PPT) (MVS Initialization and Tuning Reference).

Procedure

To update the Db2 definitions to IRLM and MVS for Db2 12, complete the following steps:

1. Examine the DSNTIJMV job and study the updates that it makes.
2. Update the following SYS1.PARMLIB members by editing them directly, or by tailoring and running job DSNTIJMV:
**IEAAPxx or PROGxx**

This member is updated to include the Db2 program libraries (`prefix.SDSNEXIT`, `prefix.SDSNLOAD`, `prefix.SDXRRESL`, and `prefix.SDSNLINK`) as libraries that are authorized by using the authorized program facility (APF).

`prefix.SDSNLINK` does not need to be included in the list of APF-authorized libraries unless LNKAUTH=APFTAB is set.

All libraries that are concatenated with `prefix.SDSNEXIT` and `prefix.SDSNLOAD` in STEPLIB and JOBLIB statements must be APF-authorized.

**LNKLSTxx**

This member is updated to include the Db2 load module library, `prefix.SDSNLINK`, in the LNKLSTxx. If you moved the modules from `prefix.SDSNLINK` into another library, edit DSNTIJMV to include that library in the LNKLSTxx. If you have combined `prefix.SDSNLINK` and `prefix.SDSNLOAD` into one library, edit DSNTIJMV to include the combined library in the LNKLSTxx.

**Tip:**

You can do additional editing for the SYS1.PARMLIB updates. If you are editing DSNTIJMV rather than making the changes directly, you have a choice: You can either include your additional entries for the SYS1.PARMLIB members (IEAAPxx and LNKLSTxx) at the end of the existing list of entries, or you can place them earlier in the list.

If you include these entries at the end of the existing SYS1.PARMLIB list, ensure that commas (the continuation character) delimit each entry except the last entry.

ECSA size is another SYS1.PARMLIB change to consider at this time. You specify ECSA size in the CSA parameter of the IEASYS00 parameter. Ensure that you have specified an adequate size for this subsystem.

3. If necessary, update the NSYSLX option of the z/OS IEASYSxx PARMLIB member to define more z/OS system linkage indexes (LXs). Each Db2 and each IRLM that you define to z/OS in the IEFSSNxx PARMLIB member requires a z/OS system linkage index (LX). The default number of these indexes that z/OS reserves is 165. If you place all of your Db2 and IRLM subsystem definitions in a single IEFSSNxx member, you might need more than 165 LXs; otherwise, your subsystems might not start.

4. Complete the following steps to enable diagnosis of IRLM problems:

   a. Specify the IRLM dump formatting modules DXRRLM50, DXRRLM55, and DXRRLS55 in a BLSCUSER member in a data set that is in the system's PARMLIB concatenation.

   b. Copy the DXRR186, DXRRFTB, DXRRRLM50, DXRRLM55, and DXRRLS55 modules from the IRLM SDXRRESL data set to a data set in the system's link list. Or, add the SDXRRESL data set to the system's link list in a PROGxx member of a data set in the system's PARMLIB concatenation. If you use more than one level of IRLM, you must use a STEPLIB concatenation when using IPCS to process dumps that are taken for the levels whose modules are not in the link list.

**Related tasks:**

- [Migration step 17: Refresh the z/OS definition of Db2](#)
Migration step 14: update Db2 address space start and language procedure definitions: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ

During migration to Db2 12, you must update the definitions of Db2 in SYS1.PROBLIB or the user-defined procedure library of the target z/OS system.

Before you begin

Ensure that the following task is complete: Make Db2 load modules available to TSO and batch users.

About this task

The jobs in this task rename your existing Db2 11 procedures so that they do not conflict with the Db2 12 procedures, which replace your old procedures.

Procedure

To update the system procedure library for the target z/OS system for migration to Db2 12, complete the following steps:

1. Update the Db2 and IRLM address space startup procedures by tailoring and running job DSNTIJMA. Job DSNTIJMA renames and replaces the startup procedures in the target system procedure library for the following Db2 address spaces, where ssnn is the subsystem name:
   • ssnnMSTR for Db2 system services
   • ssnnDBM1 for Db2 database services
   • ssnnDIST for the Db2 distributed data facility (DDF)
   • ssnnADMT for the Db2 administrative task scheduler
   • IRLMPROC, or a user-defined procedure for IRLM

Examine the size of the private area on the Db2 startup procedures. If necessary, modify the procedures to satisfy the requirements for environmental descriptor manager (EDM) pool size, buffers, number of data sets open, and amount of available private address space. For more information, see Calculating working storage requirements.

The STEPLIB concatenation of the ssnnDBM1 address space procedure includes a commented-out DD for the IBM Language Environment data sets SCEERUN and SCEERUN2. If your system does not include SCEERUN and SCEERUN2 in the system link list, you must uncomment this DD.

2. Update the address space startup procedures for the Db2-supplied WLM environments to by tailoring and running job DSNTIJMW. Job DSNTIJMW renames and replaces the following procedures in the system procedure library for WLM environments:
   • ssnnWLM for the sample WLM environment
   • ssnnWLMC for DSNWLM_DSNACICS
   • ssnnWLMD for DSNWLM_DEBUGGER
   • ssnnWLMG for DSNWLM_GENERAL
   • ssnnWLMJ for DSNWLM_JAVA
   • ssnnWLMM for DSNWLM_MQSERIES
   • ssnnWLMP for DSNWLM_PGM_CONTROL
   • ssnnWLMR for DSNWLM_REXX
3. Copy the Db2 language procedures by running job DSNTIJMP. Job DSNTIJMP renames and replaces the following Db2-supplied language procedures to the target system procedure library:

- `ssnmWLM1` for `DSNWLM_NUMTCB1`
- `ssnmWLMU` for `DSNWLM_UTILS`
- `ssnmWLMW` for `DSNWLM_WEBSERVICES`
- `ssnmWLMX` for `DSNWLM_XML`

Change any data set names that differ at your site. If you specified a data set name suffix in the DATA SET NAME SUFFIX field of DSNTIPA1, that suffix is appended to data sets &USER..DBRMLIB.DATA, &USER..RUNLIB.LOAD, and &USER..SRCLIB.DATA. To override these data set names, you must edit the updates to the target system procedure library.

The language preparation procedures in job DSNTIJMP use the DISP=OLD parameter to enforce data integrity. However, when the installation CLIST runs, the DISP=OLD parameter for the DBRM library data set is modified to DISP=SHR. This might cause data integrity problems when you run multiple precompiler jobs. To avoid these data integrity problems, if you are not using DFSMSdfp partitioned data set extended (PDSE), you must change the language preparation procedures (DSNHICOB, DSNHICB2, DSNHFOR, DSNHC, DSNHCPP, DSNHCPP2, DSNHPLI, DSNHASM, and DSNHSQL) to specify the DISP=OLD parameter instead of the DISP=SHR parameter.

If compiler STEPLIB statements are needed, add them.

4. Update the environment files for the Db2-supplied Java WLM environment by tailoring and running job DSNTIJMJ. Job DSNTIJMJ has the following three steps that update environment files for the Db2-supplied WLM environment, `DSNWLM_JAVA`:

a. Job step DSNTIJJ creates a Language Environment run-time options (JAVAENV) data set for the `ssnmWLMJ` address space procedure. This data set is allocated by the JAVAENV DD of the `ssnmWLMJ` address space procedure that is created in job DSNTIJMW.

In this job step, review the `JAVA_HOME` path, which is the path where the JDK is located, and update if necessary. If you are using a 64-bit JVM, use the following path:

```
JAVA_HOME=/usr/lpp/java160/J6.0_64
```

If you are using a 31-bit JVM, use the following path for Java 5:

```
JAVA_HOME=/usr/lpp/java150/J5.0
```

Or use the following path for Java 6:

```
JAVA_HOME=/usr/lpp/java160/J6.0
```

You cannot specify two JDKs in a single `JAVA_HOME` path.
• _CEE_ENVFILE if present, must be set to the name and path of the environment variable file that was created by job step DSNTIJR.
• JVMPROPS needs to be set to the name and path of the JVM startup options file that was created by job step DSNTIJS.
• DB2_BASE and JCC_HOME need to be set to the names of the HFS directories that you selected for the Db2 for z/OS and JCC product classes.

You cannot specify two JDKs in a single JAVA_HOME path.
b. Job step DSNTIJR creates an environment variable file in HFS.
c. Job step DSNTIJS creates a JVM startup options file in HFS.
d. The absolute path and name of a HFS or zFS file is passed to DSNTJMJX.
e. DSNTJMJX checks whether the directory where this file is located exists or not. If the directory does not exist, it is created.

5. After running job DSNTIJMJ, review the content of the Language Environment options data set and adjust the following environment variables as needed. The data set is created by DSNTIJMV and must be allocated by the JAVAENV DD of the address space procedure for the Db2 core WLM environment for Java routines.
• If _CEE_ENVFILE is present, set it to the name and path of the environment variable file that was created by job step DSNTIJR.
• Set JVMPROPS the name and path of the JVM startup options file that was created by job step DSNTIJS.
• Set DB2_BASE and JCC_HOME to the names of the HFS directories that you selected for the Db2 for z/OS and JCC product classes.

Migration step 15: Define system data sets: DSNTIJIN

When job DSNTIJIN is tailored for migration, it defines the following non-VSAM data sets: prefix.SRCLIB.DATADTA, prefix.RUNLIB.LOAD, and prefix.DBRMLIB.DATADTA.

About this task

When job DSNTIJIN is tailored for migration, it does not define data sets for new Db2 catalog and directory table spaces and indexes. Catalog and directory data sets are automatically defined by Db2 when you run job DSNTIJTC to tailor the catalog.

When migrating to Db2 12, do not use DSNTIJIN or any other job to define data sets for the Db2 catalog and directory.

Procedure

To define prefix.SRCLIB.DATADTA, prefix.RUNLIB.LOAD, and prefix.DBRMLIB.DATADTA:

Run job DSNTIJIN.

Migration step 16: Define user authorization exit routines: DSNTIJEX (optional)

You can use job DSNTIJEX to define user authorization exit routines. If you did not run job DSNTIJEX in the release from which you are migrating, you can skip this step.
About this task

Job DSNTIJEX builds sample authorization exit routines from the source code in prefix.SDSNSAMP. These sample authorization exit routines are DSN3@SGN and DSN3@ATH, and the user version of the access control authorization exit routine DSNX@XAC. DSNTIJEX then places the exit routines in the prefix.SDSNEXIT library.

Job DSNTIJEX includes a step to assemble and link-edit the sample version of DSNACICX, which you can use to modify CICS parameters that the DSNACICS caller specifies. You can modify the access control authorization exit routine DSNX@XAC and use DSNTIJEX to assemble and link-edit it. This exit routine allows you to bypass some or most of Db2 authorization checking and to specify your own authorization checking. The Db2 CLIST tailors the JCL in DSNTIJEX to meet the requirements of your site.

The sample authorization exit routines are not the same as the default authorization exit routines that are supplied by Db2. By implementing the sample authorization exit routines, you can provide group names as secondary authorization IDs.

You have the following options regarding exit routines:
• To use the default authorizations, skip job DSNTIJEX.
• To use the sample authorization exit routines, run job DSNTIJEX.
• To use your own authorization exit routines, modify job DSNTIJEX to reference the correct library, and then run it.

Procedure

If you choose to run job DSNTIJEX:
1. Make the following optional adjustments to DSNTIJEX if they apply to your site:
   • DSNXSXAC is a copy of the default access control authorization exit routine that you can modify. This exit routine allows you to bypass some or most of Db2 authorization checking and to specify your own authorization checking. If you do not change the exit routine, you should delete this step.
   • DSNACICS is a stored procedure that invokes user exit routine DSNACICX, which you can use to modify CICS parameters that the DSNACICS caller specifies. If you do not need to modify the caller's parameter values, you can use the default DSNACICX exit routine. However, if you need to modify the caller's parameter values, you need to perform the following tasks:
     a. Write a user exit routine in assembler, COBOL, C, or PL/1
     b. Assemble or compile the source code
     c. Link-edit the object code into the Db2 exit routine library
   Installation job DSNTIJEX includes a step to assemble and link-edit the sample version of DSNACICX. You can use this step as a model for your program preparation job.
   • If you will use the RACF/Db2 external security module (DSNXRXAC) as your Db2 access control authorization exit, modify job DSNTIJEX to refer to DSNXRXAC instead of DSNXSXAC.
2. Run job DSNTIJEX.
   If job DSNTIJEX runs successfully, it produces a return code of 0 or 4.
   If job DSNTIJEX fails or abends, correct the problem, and rerun the job.
Related concepts:

- Access control authorization exit routine (Managing Security)
- Db2 data access control (Managing Security)

Related information:

- Exit routines (Db2 Administration Guide)

Migration step 17: Refresh the z/OS definition of Db2

As part of the migration to Db2 12, you must refresh how Db2 is defined to z/OS.

Before you begin

Complete the following tasks:

- Ensure that the early code pre-conditioning PTFs are installed on your system before you migrate. For details, see Required maintenance for Db2 12 installation or migration.
- Determine the version of the ERLY code to include in the system link list, as described in Choosing link list options. In most cases, use the ERLY code that is distributed with the Db2 release you are migrating to. The link list can contain only one level of the early code, and it must support all Db2 subsystems and different Db2 releases that share it.
- Update the definition for Db2 12 to IRLM and MVS, as described in Migration step 13: Define Db2 12 to IRLM and MVS: DSNTIJMV.

About this task

This migration task is necessary to apply changes for Db2 12 migration in the following z/OS structures:

- The link list look-aside address space (LLA), which contains the Db2 12 early (ERLY) code from the SDSNLINK load module library.
- Member IEAAPFx, for new APF libraries.
- Member LNKLSTxx, for new load module libraries.
- JCL startup procedures for the Db2 12 address spaces, including WLM environments used by stored procedures and user-defined functions.
- JCL procedures for preparing Db2 12 application programs.

Procedure

Refresh how Db2 is defined to z/OS by using one of the following approaches:

- Do an initial program load (IPL) of z/OS. You must choose this approach if you issued z/OS SETPROG LINKLST commands to refresh the system link list in member LNKLSTxx.
  After the IPL of z/OS completes, message DSN3100I is displayed on the z/OS console, stating that Db2 is ready for the START command.
- If you changed to a new version of the ERLY code, and you did not refresh the system link list in member LNKLSTxx, run the -REFRESH DB2,EARLY command from a z/OS console that has START command capability. When -REFRESH DB2,EARLY is run, previous copies of the ERLY modules are deleted from the system the next time that Db2 is started. -REFRESH DB2,EARLY is valid only when Db2 is not started.
Migration step 18: Start Db2 12

During migration to Db2 12, start the Db2 subsystem after you run job DSNTIJMV and IPL z/OS.

Before you begin

If you have not already done so, convert the BSDS to extended format, as described in Convert the BSDS, Db2 catalog, and directory to 10-byte RBA and LRSN format.

Procedure

To start Db2:

1. Start the IRLM. If you have not requested that Db2 automatically start the IRLM, you should start it before you start Db2. Use the following command, where irlmproc is the name that you assigned to the IRLM startup procedure:

   ```
   START irlmproc
   ```

   The `irlmproc` is the value that you specified for the PROC NAME option on installation panel DSNTIPI.

   If you specified YES for the AUTO START option on installation panel DSNTIPI, Db2 starts the IRLM automatically.

2. Start Db2 from the z/OS console with the following command, where -DSN1 is the subsystem command prefix that you defined for Db2, and `DSNZPxxx` is the name of the Db2 initialization parameter module:

   ```
   -DSN1 START DB2 PARM(DSNZPxxx)
   ```

   If you omit the PARM parameter, and you specified a value in the PARAMETER MODULE field on installation panel DSNTIPO3, that value is used as the parameter module name. If you did not specify a parameter module on panel DSNTIPO3, Db2 uses the default, DSNZPARM.

   If you have not run job DSNTIJRT, and the administrative task scheduler starts, the administrative task scheduler issues error message DSNA679I to report SQLCODE -805 until you run job DSNTIJRT. The administrative task scheduler starts if you specified an address space procedure name in the ADMIN SCHEDULER field on installation panel DSNTIPX.

   If Db2 starts successfully, two to five address spaces also start. These address spaces are `ssnmMSTR` and `ssnmDBM1`, possibly `ssnmDIST`, possibly `ssnmADMT`, and `irlmproc`, where `ssnm` is the Db2 subsystem name, `ssnmADMT` is the Db2 administrative task scheduler procedure name, and `irlmproc` is the IRLM procedure name.
If Db2 starts successfully, the series of REST
AR
T messages that you receive
concludes with these two messages:

DSNR002I  RESTART COMPLETED
DSN022I  DSNYASCP 'DSN1 START DB2' NORMAL COMPLETION

In the next migration step, you migrate the Db2 catalog. Before the catalog is
migrated, some catalog or directory table spaces are restricted.

The following messages might occur during startup of Db2 because the catalog
and directory table spaces are restricted:

• DSN500I with reason code 00C900A6
• DSN501I with reason code 00C900A6
• DSNL700I with reason code 00C900A6 (if DDF is auto-started)
• Abend 04E with reason code 00E70014 (during DDL registration)

These restrictions can cause failure of processes that are started as part of
normal Db2 startup, such as collection of real-time statistics, until catalog
tailoring is completed. Possible actions to restart such processes are explained
in the discussion of job DSNTIJTC.

You can determine existing restrictions by issuing this command after you start
Db2:

-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT

The preceding command might also generate message DSN501I with reason
code 00C900A6.

If Db2 does not start properly, it usually abends with a reason code that
indicates where the error occurred. To find the error, check the set of definitions
for the associated resource. A common cause of startup failure is that the BSDS
does not match the subsystem parameter values; ensure that the correct job was
run for DSNTIJUZ. Also, check that the subsystem parameter member that you
specified (or used by default) when you started Db2 is the one that the
DSNTIJUZ job built. Check the JCL for the Db2 startup procedure.

3. If you want to use the TSO SUBMIT command to do housekeeping and
migration verification, start TSO (if it is not already started).

4. If you did not enable primary user IDs to issue Db2 commands from the z/OS
console or TSO SDSF when you first installed the subsystem, take one of the
following actions:

• Grant SYSOPR authority to all primary user IDs that issue Db2 commands
  from the z/OS console or TSO SDSF. Issue the following command:
    GRANT SYSOPR TO userid

• Define RACF classes to authorize Db2 commands. Use the following
  statements:
    SETR CLASSACT(DSNADM)
    RDEFINE DSNADM DSN1.SYSOPR UACC(NONE)
    SETR RACLIST(DSNADM) REFRESH
    PERMIT DSN1.SYSOPR CLASS(DSNADM) ID(userid) ACCESS(READ)
    SETR RACLIST(DSNADM) REFRESH

Related reference:

-START DB2 (Db2) (Db2 Commands)
-DISPLAY DATABASE (Db2) (Db2 Commands)

Related information:

DSNJ157I (Db2 Messages)
Migration step 19: Tailor Db2 12 catalog: DSNTIJTC

Job DSNTIJTC invokes the CATMAINT utility to migrate the catalog of the Db2 11 to the Db2 12 catalog.

Before you begin

About this task

DSNTIJTC contains the following job steps:

- Job step DSNTITC, which creates new catalog and directory objects, adds columns to existing catalog tables, and creates and updates indexes on the catalog tables to accommodate new Db2 12 objects. All IBM-supplied indexes are created or updated sequentially during the execution of this step.
- Job step DSNTIRI, which executes only when job step DSNTITC terminates abnormally or with a return code of 8 or greater. DSNTIRI rebuilds catalog indexes that were placed in REBUILD-pending status during execution of DSNTITC.

Procedure

To complete the tailoring of the Db2 catalog:

1. Run job DSNTIJTC on your Db2 12 subsystem.
   
   A status message, DSNU777I, is issued at several points to indicate migration progress. Diagnostic error messages are issued when CATMAINT processing fails. If a problem is found during the SQL processing phase of migration, message DSNU778I is issued. If non-supported functions are encountered, message DSNU776I is issued. All of these messages are written to the SYSPRINT data set.
   
   If you receive abend reason code 00C200EF, the catalog and directory data sets are not in a compliant SMS environment.

2. If job DSNTIJTC fails, save the output and correct the problem.
   
   a. Terminate the CATMAINT utility if it is pending.
   
   b. Run job DSNTIJTC again without modification.

3. If job DSNTIJTC fails again, return to your previous Db2 release. Because CATMAINT failures roll back all Db2 12 changes, the catalog and directory are in the format of your previous Db2 release. Altered indexes are not rolled back. Determine if any index is in a pending status by using the CHECK INDEX utility. To return to your previous Db2 release:
   
   a. Rename procedures to use the libraries of your previous release.
   
   b. Reconnect TSO, IMS, and CICS to the libraries of your previous release.

4. If your previous release is damaged, you need to complete a point-in-time recovery to recover the system. To complete a point-in-time recovery:
   
   a. Restore your previous version’s catalog and directory from image copies.
   
   b. Restore the BSDSs from archive logs that were made prior to migration.
   
   c. Flush the SCA (for data sharing environments only).
   
   d. Recover the catalog and directory indexes.
   
   When you remigrate to Db2 12, run the DSNTIJTC job again.
Results

When you start Db2 12 for the first time, you might receive message DSNT501I with reason code 00C900A6. You can ignore this message. Running job DSNTIJTC corrects the cause of this message.

The catalog is now tailored at catalog level V12R1M500, which supports Db2 12 function level 100 - function level 501. For more information about function levels and catalog levels in Db2 12 continuous delivery, see Function levels and related levels in Db2 12 (Db2 for z/OS What's New?).

Related concepts:
- Work file database storage requirements

Related tasks:
- Falling back to Db2 11

Related reference:
- CATMAINT (Db2 Utilities)

Related information:
- DSNU776I (Db2 Messages)
- DSNU777I (Db2 Messages)
- DSNU778I (Db2 Messages)

Migration step 20: Ensure that the catalog has no problems (optional)

After you tailor your Db2 12 for z/OS catalog, you should check the integrity of your catalog and directory.

Procedure

To check the integrity of your catalog and directory, complete these steps in any order:

- Run the CHECK INDEX utilities on all the indexes in the catalog and directory by using job DSNTIJCX.
- Run the queries in member DSNTESQ of prefix.SDSNSAMP. Because SPUFI is not yet bound, you cannot use SPUFI to run these queries. One alternative is to use the Db2 11 DSNTEP2 program to run the queries.
- Run the DSN1COPY utility with the CHECK option on the catalog table spaces.

Related tasks:
- Migration step 2: Verify the integrity of Db2 table spaces (optional)

Related reference:
- DSN1COPY (Db2 Utilities)

Migration step 21: Prepare dynamic SQL program: DSNTIJTM

Job DSNTIJTM assembles, link-edits, binds, and runs DSNTIAD, a program that processes certain SQL statements dynamically.
Procedure

To prepare the dynamic SQL program:

1. Review and possibly modify or delete the GRANT statements in job DSNTIJTM. This job grants PUBLIC access to a number of Db2-supplied objects. Consider granting execute access on the plan to PUBLIC.

2. Run job DSNTIJTM.

Migration step 22: Bind SPUFI and DCLGEN and user-maintained database activity: DSNTIJSG

Use job DSNTIJSG to bind packages for SPUFI, DCLGEN, and the EXEC SQL utility.

About this task

In migration mode, DSNTIJSG completes the following actions:

- Binds packages at the new release level for SPUFI, DCLGEN, and the EXEC SQL utility.
- Creates a new resource limit facility table, DSNRLMTnn, and a new resource limit facility index, DSNMRLnn. By default, the step that creates this new table and index is commented out in the JCL. To have Db2 create DSNRLMTnn and DSNMRLnn, uncomment the step.
- Adds new EXPLAIN tables under the SYSIBM schema.

Procedure

To bind the SPUFI, DCLGEN, and the EXEC SQL packages for Db2 12, complete the following steps:

1. If you bound special SPUFI packages and plans in Db2 11, you need to bind those packages again in Db2 12. You do not need to bind those plans again. For example, to update special SPUFI packages that were created for use by SPUFI users who require a TSO terminal CCSID of 1047, issue the following commands.

```sql
BIND PACKAGE(SPUR1047) MEMBER(DSNESM68) -
  ACTION(REPLACE) ISOLATION(UR) ENCODING(1047) -
  LIBRARY('prefix.SDSNDBRM')
BIND PACKAGE(TIAP1047) MEMBER(DSNTIAP) -
  ACTION(REPLACE) ISOLATION(CS) ENCODING(1047) -
  LIBRARY('prefix.SDSNDBRM')
BIND PACKAGE(SPCS1047) MEMBER(DSNESM68) -
  ACTION(REPLACE) ISOLATION(CS) ENCODING(1047) -
  LIBRARY('prefix.SDSNDBRM')
BIND PACKAGE(SPRR1047) MEMBER(DSNESM68) -
  ACTION(REPLACE) ISOLATION(RR) ENCODING(1047) -
  LIBRARY('prefix.SDSNDBRM')
```

2. If you are using a data sharing environment, you must ensure that the resource limit facility (RLF) is inactive on all members in the data sharing group before running DSNTIJSG. To do this, issue the STOP RLIMIT command for each member.
3. Ensure that all your subsystems have the same maintenance level for your DSNUGSQL DBRM.

4. Run job DSNTIJSG. If DSNTIJSG runs successfully, it produces a return code of 0. However, job step DSNTIJG can produce a return code of 4 if the specified access has already been granted on one or more objects.

5. If job DSNTIJSG fails or abends, ensure that the user that is specified on the JOB statement is authorized. Use the name that you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP1. (The RESTART parameter on the JOB statement can be useful.) Correct any other problems, and run job DSNTIJSG again.

6. If you encounter resource shortages, review the parameters in job DSNTIJUZ and make any necessary modifications. Then, stop Db2, rerun DSNTIJUZ, start Db2, and rerun DSNTIJSG from the last successful step.

**What to do next**

Complete the following actions:

- Alter your Db2 12 buffer pools that have frequent page reads or frequent page writes to use PGFIX YES if you have sufficient real storage available for these buffer pools. Fixing the buffer pages in real storage once and keeping them fixed avoids the processing time that Db2 needs to fix and free pages each time there is an I/O. In some cases, this processing time can be as much as 10% for I/O-intensive workloads. To use this option, issue the following command:

  ALTER BPOOL(bpname) VPSIZE(vpsize) PGFIX(YES)

Where *bpname* is the name of the buffer pool and *vpsize* is the size of the virtual pool.

- If you use capabilities such as ODBC, JDBC, and others with Db2, you might need to run other jobs to migrate them for use with Db2 12, for example:
  - For ODBC, see Migrating to the Db2 12 ODBC driver (Db2 Programming for ODBC).
  - For JDBC see Upgrading the IBM Data Server Driver for JDBC and SQLJ to a new version (Db2 Application Programming for Java).

**Related tasks:**

- Migration step 29: Activate function level 500
- Special packages and plans for SPUFI

**Migration step 23: Configure Db2 for running stored procedures and user-defined functions (optional)**

This step needs to be performed during installation, and is therefore optional for migration. Before you can use user-defined functions or the following types of stored procedures, you need to perform some set-up tasks:

- External stored procedures
- External SQL procedures
- Native SQL procedures that satisfy at least one of the following conditions:
The native SQL procedure calls at least one external stored procedure, external SQL procedure, or user-defined function. The native SQL procedure is defined with ALLOW DEBUG MODE or DISALLOW DEBUG MODE.

- Db2-supplied stored procedures

All routines, except native SQL procedures, must run in an environment that is established by the z/OS Workload Manager (WLM). A set of Db2-supplied routines are installed by job DSNTIJRT and validated by job DSNTIJRV as part of the migration process. You can also define your own routines or use routines that are provided by a non-IBM vendor. Before you can use any stored procedures and user-defined functions, however, you need to set up WLM environments for these routines. If you plan to use certain Db2-supplied stored procedures or user-defined functions, you need to perform some additional steps.

Related concepts:
- Stored procedures (Db2 Application programming and SQL)
- User-defined functions (Db2 SQL)

Tailoring Db2 subsystem parameters for stored procedures during migration (optional)
Several subsystem parameters should be tailored to your Db2 environment before you use stored procedures. These parameters were set when you installed or migrated Db2. If you are satisfied with those values, you do not need to complete this step.

About this task

The subsystem parameters for stored procedures are:
- MAX_NUM_CUR
- MAX_ST_PROC
- STORMXAB
- STORTIME

Procedure

To modify the values of these subsystem parameters, complete one of the following tasks:
- Run the installation CLIST again in INSTALL or MIGRATE mode. On the installation panels, leave most of the existing values and change the following fields on their respective panels:
  1. On panel DSNTIP1, specify the input member that contains field values for your current installation.
  2. On panel DSNTIP5, specify different names for the TEMP CLIST LIBRARY and SAMPLE LIBRARY fields to avoid overwriting your original libraries.
  3. On panel DSNTIPX, add or change the values of the MAX ABEND COUNT, TIMEOUT VALUE, MAX OPEN CURSORS, and MAX STORED PROCS fields as appropriate.
  4. On panel DSNTIPY, specify a name for the REMOTE LOCATION field. Db2 uses this name for the stored procedure sample applications.
- Edit job DSNTIJUZ to add or change values for the subsystem parameters for routines.
This method has the following disadvantages:

– Db2 does not use these new values for any future migrations because your DSNTIDxx member and DSNTIJUZ parameters are not saved for future input.

– Db2 does not generate the sample jobs for stored procedure and user-defined functions because you are not running the DSNTINST CLIST.

Related reference:

- DSNTIPA1 or DSNTILA1: Main panel
- DSNTIPT: Db2 data set names panel
- DSNTIPX: Routine parameters panel
- DSNTIPY: Job editing panel
- Directory of subsystem parameters, panel fields, and application default values

Setting up Language Environment for stored procedures during migration

Language Environment (LE) is a component of the z/OS operating system that establishes a common run time environment for all applications, regardless of programming language. Db2 uses Language Environment to provide a run time environment for external stored procedures and external SQL procedures.

About this task

Language Environment provides a set of essential run time services that are available through a set of interfaces that are consistent across programming languages. These services include routines for run time message handling, condition handling, and storage management.

Language Environment provides the following advantages for Db2 stored procedures:

- Language Environment hides the differences between programming languages.
- Language Environment provides the ability to make a stored procedure resident in the stored procedure address spaces.
- Language Environment supports a large number of run time options, including those options that are needed to use tools to debug your stored procedures.

Procedure

To set up Language Environment for stored procedures:

1. Customize Language Environment by completing the instructions in z/OS Language Environment Customization

2. Specify the Language Environment run time library in the JCL startup procedure for each WLM-managed address space for stored procedures. This library must be the only available run time library. Do not reference any other language run time libraries within the system link list or within the STEPLIB statements. If other language run time libraries are defined in the system link list, use STEPLIB statement overrides to exclude them.
   Multiple languages can share the same Language Environment run time library. Therefore, you do not need to specify the language-specific libraries in the JCL startup procedure of each stored procedure address space.

Restriction: Do not use compilers (or associated run time libraries) that are no longer supported by IBM.
What to do next

After you set up Language Environment for stored procedures, you can override the Language Environment run time options for individual stored procedures. You can do this by using the RUN OPTIONS clause in the CREATE PROCEDURE statement or ALTER PROCEDURE statement. The default values for Language Environment run time options are defined by z/OS. You might have overridden some of these default values when you customized Language Environment.

Related concepts:
- External stored procedures (Db2 Application programming and SQL)
- SQL procedures (Db2 Application programming and SQL)

Related reference:
- CREATE PROCEDURE (external) (Db2 SQL)
- CREATE PROCEDURE (SQL - external) (Db2 SQL)
- Customizing Language Environment run-time options (z/OS Language Environment Customization)

Related information:
- Implementing RRS for stored procedures during migration

Implementing RRS for stored procedures during migration
To use some Db2 stored procedures, you must implement Resource Recovery Services (RRS). RRS is a sync point manager program that coordinates two-phase commit processing. Db2 requires that RRS be active because WLM-managed stored procedure address spaces use the RRS attachment facility (RRSAF).

About this task

RRS records events for protected resources in log streams that are managed by a z/OS component called the system logger. The following table identifies the log streams that RRS uses, where gname can be your Sysplex name or any name in a non-Sysplex environment:

Table 47. Log streams that RRS uses

<table>
<thead>
<tr>
<th>Log stream type</th>
<th>Log stream name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main unit-of-recovery log state stream</td>
<td>ATR.gname.MAIN.UR</td>
<td>The state of active units of recover (URs). RRS periodically moves this information into the RRS delayed UR state log when UR completion is delayed.</td>
</tr>
<tr>
<td>Delayed unit-of-recovery log state stream</td>
<td>ATR.gname.DELAYED.UR</td>
<td>The state of active URs, when UR completion is delayed.</td>
</tr>
<tr>
<td>Resource manager data log stream</td>
<td>ATR.gname.RM.DATA</td>
<td>Information about the resource managers that use RRS services.</td>
</tr>
</tbody>
</table>
### Table 47. Log streams that RRS uses (continued)

<table>
<thead>
<tr>
<th>Log stream type</th>
<th>Log stream name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restart log stream</td>
<td>ATR.gname.RESTART</td>
<td>Information, which is needed during restart, about incomplete URs. This information enables a functioning RRS instance to assume remaining incomplete work from an RRS instance that failed.</td>
</tr>
<tr>
<td>Archive log stream</td>
<td>ATR.gname.ARCHIVE</td>
<td>Information about completed URs. This log stream is optional, but recommended.</td>
</tr>
</tbody>
</table>

All of these log streams are required except for the archive log stream. RRS does not start if it is not able to connect to the required log streams.

In a Sysplex environment, these log streams are shared by the systems of the Sysplex.

### Procedure

To implement RRS for stored procedures:

1. Define the RRS log streams that are to be written, either on disk or in the coupling facility by completing the following actions:
   
a. If you are using the coupling facility, add definitions for the RRS structures in the CFRM policy.

   When you are use coupling facility log streams, each log stream needs to be mapped to a coupling facility structure.

**Example:** The following example shows JCL that updates the CFRM policy to add the RRS structures:

```
//DEFCFRM1 JOB MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*  
//SYSABEND DD SYSOUT=*  
//SYSIN DD *
  DATA TYPE(CFRM) REPORT(YES)  
  DEFINE POLICY NAME(CFRM18) REPLACE(YES) 
  CF NAME(CF01)  
    TYPE(009672)  
    MFG(IBM)  
    PLANT(02)  
    SEQUENCE(000000040104)  
    PARTITION(1)  
    CPCID(00)  
    DUMPSPACE(2048)  
  CF NAME(CF02)  
    TYPE(009672)  
    MFG(IBM)  
    PLANT(02)  
  
  STRUCTURE NAME(RRS_ARCHIVE_1)  
    INITSIZE(8000)  
    SIZE(16000)  
    PREFLIST(CF1,CF2)  
```
b. Define the log streams in the system logger policy by using the IXCMIAPU utility. This utility program is provided in the SYS1.MIGLIB system library.

If you are using the coupling facility, also define the structures in the system logger policy. You can map each log stream to a single structure or you can map log streams of like data types to the same structure.

Example of defining coupling facility log streams: The following example JCL shows definitions for coupling facility structures and RRS log streams. In this case, each RRS log stream is mapped to a structure.

```jcl
//STEP1_EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
DATA TYPE(LOGR) REPORT(YES)

DEFINE STRUCTURE NAME(RRS_ARCHIVE_1) LOGSNUM(1)  
MAXBUFSIZE(64000) AVGBUFSIZE(262)

DEFINE STRUCTURE NAME(RRS_RMDATA_1) LOGSNUM(1)  
MAXBUFSIZE(1024) AVGBUFSIZE(252)

DEFINE STRUCTURE NAME(RRS_MAINUR_1) LOGSNUM(1)  
MAXBUFSIZE(64000) AVGBUFSIZE(158)

DEFINE STRUCTURE NAME(RRS_DELAYEDUR_1) LOGSNUM(1)  
MAXBUFSIZE(64000) AVGBUFSIZE(158)

DEFINE STRUCTURE NAME(RRS_RESTART_1) LOGSNUM(1)  
MAXBUFSIZE(64000) AVGBUFSIZE(158)

DEFINE LOGSTREAM
NAME(ATR.SANDBOX.ARCHIVE) STRUCTNAME(RRS_ARCHIVE_1)  
LS_DATACLAS(SHARE33)  
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)  
LOWOFFLOAD(0) HIGHTOFFLOAD(80) STG_DUPLEX(NO)  
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM
NAME(ATR.SANDBOX.RM.DATA) STRUCTNAME(RRS_RMDATA_1)  
LS_DATACLAS(SHARE33)  
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)  
LOWOFFLOAD(0) HIGHTOFFLOAD(80) STG_DUPLEX(NO)  
RETPD(15) AUTODELETE(YES)
```
DEFINE LOGSTREAM
NAME(ATR.SANDBOX.MAIN.UR) STRUCTNAME(RRS_MAINUR_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM
NAME(ATR.SANDBOX.DELAYED.UR) STRUCTNAME(RRS_DELAYEDUR_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

DEFINE LOGSTREAM
NAME(ATR.SANDBOX.RESTART) STRUCTNAME(RRS_RESTART_1)
LS_DATACLAS(SHARE33)
HLQ(LOGR) MODEL(NO) LS_SIZE(1024)
LOWOFFLOAD(0) HIGHOFFLOAD(80) STG_DUPLEX(NO)
RETPD(15) AUTODELETE(YES)

/*
Example of defining disk log streams: The following example JCL shows
definitions for RRS log streams on disk.
DEFSTREXECPGM=IXCMIAPU
//SYSPRINT DD(SYSOUT=A)
//SYSIN DD *
DATA TYPE(LOGR)

DEFINE LOGSTREAM NAME(ATR.PLEX1.MAIN.UR)
LOWOFFLOAD(60)
HIGHOFFLOAD(80)
DASDONLY(YES)
HLQ(TEST)
LS_SIZE(1024)
LS_DATACLAS(VSAMLS)
STG_SIZE(1024)

DEFINE LOGSTREAM NAME(ATR.PLEX1.DELAYED.UR)
LOWOFFLOAD(60)
HIGHOFFLOAD(80)
DASDONLY(YES)
HLQ(TEST)
LS_SIZE(960)
LS_DATACLAS(VSAMLS)
STG_SIZE(960)

DEFINE LOGSTREAM NAME(ATR.PLEX1.ARCHIVE)
LOWOFFLOAD(0)
HIGHOFFLOAD(80)
DASDONLY(YES)
HLQ(TEST)
LS_SIZE(960)
LS_DATACLAS(VSAMLS)
AUTODELETE(YES)
RETPD(2)
STG_SIZE(2000)

DEFINE LOGSTREAM NAME(ATR.PLEX1.RM.DATA)
LOWOFFLOAD(60)
HIGHOFFLOAD(80)
DASDONLY(YES)
HLQ(TEST)
LS_SIZE(192)
LS_DATACLAS(VSAMLS)
STG_SIZE(192)
DEFINE LOGSTREAM NAME(ATR.PLEX1.RESTART)
  LOWOFFLOAD(60)
  HIGHOFFLOAD(80)
  DASDONLY(YES)
  HLQ(TEST)
  LS_SIZE(960)
  LS_DATACLASS(VSAMLS)
  STG_SIZE(960)

c. If you are using coupling facility log streams, compile and link-edit the
   changed CFRM policy.

d. If you are using coupling facility log streams, activate the updated CFRM
   policy in your Sysplex by using the following operator command:
   SETXCF START, POLICY, TYPE=CFRM, POLNAME=policyname

2. Establish the priority for RRS within the WLM policy.
   The RRS priority needs to be equal to or higher than the dispatching priority of
   Db2. You can use the SYSSTC service class for the RRS address space to achieve
   a higher dispatching priority.

3. Move the ATRRRS procedure that is supplied by IBM from SYS1.SAMPLIB to
   your SYS1.PROCLIB as member RRS, and make any necessary changes. This
   JCL procedure is used to start RRS.
   You can use a member name other than RRS, but the first four characters of
   this name must match the subsystem name that you specify in the IEFSSNx member of SYS1.PARMLIB in the next step.
   The GNAME value in the procedure must match the gname that you specified
   when defining the log streams. The default value for gname at RRS startup is
   the Sysplex name.

   Example: The following example shows a sample procedure to start RRS.
   //RRS PROC GNAME='',CTMEM='' 
   //RRS EXEC PGM=ATRIMIKE,REGION=0M,TIME=NOLIMIT, 
   // PARM='GNAME=&GNAME 
   //

4. Define RRS as a subsystem to z/OS by editing the active IEFSSNx member of
   SYS1.PARMLIB to include the following entry:
   SUBSYS SUBNAME(RRS) /* RESOURCE RECOVERY SERVICES */
   The subsystem name can be RRS or any other name that you choose. However,
   the first four characters (or fewer characters if the name is shorter) must match
   the name of the JCL procedure to start RRS.

5. Start RRS by using the following operator command:
   START RRS
   If you subsequently need to stop RRS, use one of the following operator
   commands:
   • SETRRS CANCEL
   • SETRRS SHUTDOWN
   If neither of these commands stops RRS, use the FORCE RRS, ARM command. In
   this command, RRS is the subsystem name that your installation assigned to
   RRS in PARMLIB member IEFSSNx.

Related reference:

- z/OS MVS Programming: Resource Recovery
- Configuring and defining RRS logging requirements (z/OS MVS Programming: Resource Recovery)
Setting up a WLM application environment for stored procedures during migration

All stored procedures and user-defined functions (except native SQL procedures) must run in WLM-managed stored procedure address spaces. You must assign each stored procedure to a WLM application environment, which routes the stored procedure work to the appropriate WLM-managed address space.

About this task

If you use data sharing, see “Special considerations for WLM-managed stored procedures in coexistence” on page 704 for additional guidance about setting up WLM environments.

Recommendation: Use partitioned data set extended (PDSE) for load libraries that contain external modules for stored procedures. Using PDSEs might eliminate your need to stop and start the stored procedures address space due to growth of the load libraries. If a load library grows from additions or replacements, the library might need to be extended.

Procedure

To set up a WLM application environment for one or more stored procedures:

1. Gather requirements for the WLM environment. These requirements include:
   - The name of the Db2 subsystem or data sharing group where the stored procedure or user-defined function is to run.
   - The maximum number of task control blocks (TCBs) that can run concurrently in an address space that is initiated by the WLM environment. This setting, referred to as the NUMTCB, is the maximum number of procedures or functions that should run concurrently in that WLM-managed address space. The appropriate setting depends on the requirements or limitations of the individual stored procedure or user-defined function.
     - For recommended NUMTCB values for Db2-supplied stored procedures, see “Installation information for procedures and functions that are supplied with Db2” on page 829.
     - For recommended NUMTCB values for your own stored procedures, see Assigning stored procedures and functions to WLM application environments (Db2 Performance). If a routine allocates a large amount of memory or if the external module for the routine is not re-entrant, the NUMTCB value should be 1. If there is a need for serialization, the NUMTCB value should also be 1.
   - The WLM environment for a REXX stored procedure must have a NUMTCB setting of 1.

   • Requirements for Authorized Program Facility (APF).
     - Stored procedures and user-defined functions that execute privileged instructions (for example, WLM_REFRESH) must run in a WLM environment that has an APF-authorized address space.
Db2 REXX stored procedures, such as DSNTPSMP, and other routines that use TSO/E and REXX facilities must run in a WLM environment that does not have an APF-authorized address space.

Most stored procedures can run in a WLM environment that either has or does not have an APF-authorized address space. However, running a routine in an APF-authorized environment is recommended only when doing so is necessary.

- Special data sets and DD names that are needed by the stored procedure or user-defined function. For example, all Java stored procedures implicitly require a Language Environment run time options data set to be allocated to DD name JAVAENV. Other routines might also have an explicit requirement for a work file to be allocated to a particular DD name.

2. Create a JCL startup procedure for the WLM application environment.

Use $ssnm$WLM (where $ssnm$ is the Db2 subsystem ID) as the basis for creating the new startup procedure. $ssnm$WLM is the default name of the sample JCL procedure for starting WLM-established address spaces. If you specified a name in the WLM PROC NAME field on the DSNTIPX installation panel, the sample procedure has that specified name. The sample procedure is placed in your system PROCLIB by installation job DSNTIJMV. For the DSNCACICS stored procedure, DSNTIJMV also provides sample procedure $ssnm$CICS, where $ssnm$ is the subsystem ID. Tailor your new startup procedure as follows:

- Change the APPLENV value to the name of the WLM application environment that you plan to define.
- Change the startup procedure name to an appropriate procedure name.

**Recommendation:** Specify the same name for the application environment name and the procedure name. Using the same name simplifies monitoring the WLM environment definitions.

- Change the value of DB2SSN to your Db2 subsystem name.
- Ensure that the NUMTCB value is appropriate. If the minimum number of address spaces is specified, the NUMTCB value will be the number of task control blocks that are started in an address space.

**Recommendation:** If you want to be able to easily change the NUMTCB value, specify the NUMTCB value here, and do not specify this parameter on the application environment definition panel when you define the WLM application environment. Specifying NUMTCB on the environment definition panel associates this parameter with the WLM application environment definition. Any changes to the JCL startup procedure can be made by refreshing the WLM application environment. However, changes to the WLM application environment require that you reinstall the z/OS service policy at an LPAR or sysplex level. If you specify the NUMTCB parameter for both the WLM application environment and the startup procedure, the value in the WLM application environment overrides the value in the startup procedure.

- If the environment is one in which the cost of starting new address spaces is very high, you can use the MNSPAS parameter. The MNSPAS parameter specifies the minimum number of address spaces that are to be started and maintained. The value can be 0 through 50. The default value is 0, which means that no minimum number of address spaces is specified. If the number of address spaces is at the specified minimum, cancelling an address space causes WLM to start another one to maintain the minimum. If you
refresh a WLM environment, WLM starts a number of new address spaces as needed (or the specified minimum number) and stops the old address spaces.

If the MNSPAS parameter is specified, the NUMTCB value is the number of task control blocks that are started in an address space. If the MNSPAS parameter is not specified, the NUMTCB value is the maximum number of task control blocks that can be started in an address space.

- Edit the data set names to match your data set names.
- Ensure that the library in which the stored procedure resides is in the STEPLIB concatenation.
- Ensure that the IBM Language Environment run time library is specified in the STEPLIB concatenation, unless your system includes the Language Environment run time library in the system link list.
- If you want to run the stored procedure in a WLM environment that has an APF-authorized address space, ensure that all the libraries in the STEPLIB concatenation are defined to the Authorized Program Facility.
- Add any special DD statements that are required for the stored procedures or user-defined functions that are to run in the WLM environment.
  - If you plan to use stored procedures that use the ODBA interface to access IMS databases, make the following additional changes:
    - Add the data set name of the IMS data set that contains the ODBA callable interface code to the end of the STEPLIB concatenation. The name of this data set is usually IMS.RESLIB.
    - After the STEPLIB DD statement, add a DFSRESLB DD statement that names the IMS data set that contains the ODBA callable interface code.
  - If your routines are defined as PROGRAM TYPE SUB and use UNIX System Services (USS) syscall statements, you might need to add a NOUNDUB DD statement. By default, the USS process and its resources are cleaned up after every stored procedure and user-defined function invocation. If you need the task in this WLM environment to retain the process across subsequent invocations, add a dummy NOUNDUB DD statement. This option specifies that no cleanup will occur unless the WLM environment is refreshed. To enable this option, add the following statement to the WLM stored procedure address space startup JCL:
    //NOUNDUB DD DUMMY

Use this option with caution because routines use the USS process that was used by the previous routine that was run under the same task. Also note that any open resources such as file descriptors remain open for the next routine that is run under the same task. Consider using this option on new workloads only if reusing the USS process is the desired behavior.

**Example of a startup procedure:** The following example shows a startup procedure for a WLM application environment for external SQL, COBOL, C, C++, PL/I, and assembler stored procedures. This startup procedure for external SQL stored procedures needs one unauthorized data set included in STEPLIB.

```
//**************************************************************************
// JCL FOR RUNNING THE WLM-ESTABLISHED STORED PROCEDURES
// ADDRESS SPACE
// RGN -- THE MVS REGION SIZE FOR THE ADDRESS SPACE.
// DB2SSN -- THE DB2 SUBSYSTEM NAME.
// NUMTCB -- THE NUMBER OF TCBS USED TO PROCESS
// APPLENV -- THE MVS WLM APPLICATION ENVIRONMENT
```
3. Define the application environment to WLM by using one of the following methods.
   - Use the WLM administrative application (IWMARIN0) to define a new environment or copy or modify an existing one.
     a. In the first WLM panel, Definition Menu, enter 9 for Application Environments.
        The application environment definition panel is displayed.
     b. On the application environment definition panel, specify the appropriate action code:
        – Create a new WLM application environment.
        – Copy an existing WLM application environment.
        – Modify an existing WLM application environment.
        The appropriate panel for creating, copying or modifying the application environment is displayed.
     c. If you are creating or modifying a WLM environment, specify the appropriate options for the WLM application environment definition, using the same application environment name and procedure name that you specified for the startup procedure.
     d. After you specify the appropriate options, press Enter to create the application environment definition.
   - Modify and run sample job DSNTIJRW. As provided, DSNTIJRW creates the core Db2 WLM environments, but you can modify the job to define your own WLM environments. Job DSNTIJMV copies an address space procedure for each of the Db2 core WLM environments into SYS1.PROCLIB.

4. Install the WLM service definition that contains information about the application environment by opening the WLM Definition Menu panel and selecting the Install function in the Utilities menu bar.

5. Activate the WLM service policy from the installed service definition by completing one of the following actions:
   - Issue the VARY WLM command as follows:
     \[ \text{VARY WLM, POLICY=\text{policy-name}} \]
   - On the WLM Definition Menu panel, choose the Activate service policy option from Utilities menu bar.

6. Set up security for the WLM application environments and for creating stored procedures by completing the following actions:
   a. Optional: Specify which WLM-established address spaces can run stored procedures. If you do not complete this step, any WLM-established address space can run stored procedures.
   b. Grant access to users to create procedures in specific WLM address spaces.
   c. Grant access to users to create procedures in specific schemas. Use the GRANT statement with the CREATIN option for the appropriate schema.
d. Grant access to users to create packages for procedures in specific collections. Use the GRANT statement with the CREATE option for the appropriate collection.

**Requirement:** For native SQL procedures, the collection name for the package must be the same as the schema name.

e. Grant access to refresh the WLM environments to the appropriate people.

For more detail, see [Managing authorization for stored procedures (Managing Security)](#)

7. Ensure that the ID that is used to run the JCL startup procedure for the WLM application environment has permission to use RRSAF. Each time one of the Db2 WLM address spaces is started, it uses RRSAF to attach to Db2.

**What to do next**

When you create or alter a stored procedure, specify the WLM application environment that Db2 is to use in the WLM_ENVIRONMENT option of the CREATE or ALTER PROCEDURE statement. When you specify environments, be aware that:

- Routines that have low memory requirements and that are frequently called should not share an environment with routines that have high memory requirements and are rarely called. Otherwise, storage is wasted.
- Stored procedures can be grouped according to the authorization settings for a WLM environment.
- WLM environments should not be shared among routines that use the same DD name for different purposes.
- WLM environments should not be shared among routines if one routine allocates data sets in the environment and that routine is run only occasionally. If other routines that do not use the data sets use the environment, the data sets are kept artificially allocated, which might interfere with other tasks.

**Related concepts:**
- WLM-established stored procedures address spaces storage requirements

**Related tasks:**
- [Managing authorization for stored procedures (Managing Security)](#)
- [Refreshing WLM application environments for stored procedures (Db2 Administration Guide)](#)

**Related reference:**
- Core WLM environments for Db2-supplied routines
- DSNTWLMB tool
- [GRANT (collection privileges) (Db2 SQL)](#)
- [GRANT (schema privileges) (Db2 SQL)](#)
- [set_dub_default (BPX1SDD, BPX4SDD) (z/OS UNIX System Services Programming): Assembler Callable Services Reference](#)

**Related information:**
- [Using Operator Commands for Application Environments (z/OS MVS Planning: Workload Management)](#)
Migration step 24: Set up Db2-supplied routines

Installation jobs are provided for the setup of all Db2-supplied routines and supporting objects that are provided with the Db2 base product.

Related tasks:
- Disabling Db2-supplied routines for validation

Related reference:
- Installation information for procedures and functions that are supplied with Db2

Installing Db2-supplied routines during migration

The DSNTIJRT and DSNTIJRV jobs install and validate the installation of routines that are supplied with Db2. These jobs are configured with the options that you specified on installation panel DSNTIPR1 and panels DSNTIPRA through DSNTIPRP.

Before you begin

If the RACF facility class is active and a profile was defined for MVSADMIN.WLM.POLICY, the user that runs the DSNTIJRV job requires READ access. To grant READ access to a user, an administrator must submit the following command:

PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY) ID(user) ACCESS(READ)

If the RACF OPERCMDS class is active and a profile was defined for MVS.MCSOPER.*, the user ID that runs the DSNTIJRV job requires READ access. Alternatively, you can add a discrete profile for the DSNTRVFY console that is used in this job. To add a discrete profile for the DSNTRVFY console, an administrator must complete the following steps:

1. Establish a discrete profile for MVS.MCSOPER.DSNTRVFY by issuing the following command:
   RDEFINE OPERCMDS MVS.MCSOPER.DSNTRVFY UACC(NONE)

2. Grant READ access to the user that runs the DSNTIJRV job by issuing the following command:
   PERMIT MVS.MCSOPER.DSNTRVFY CLASS(OPERCMDS) ID(user) ACCESS(READ)

3. Refresh the OPERCMDS class by issuing the following command:
   SETROPTS RACLIST(OPERCMDS) REFRESH

Resource Recovery Services (RRS) is also required for these jobs.

Important: Job DSNTIJRT defines, binds, and grants access to stored procedures and user-defined functions. The default setting for GRANT EXECUTE is PUBLIC, meaning that anyone who can access Db2 can use these stored procedures. Use the GRANT EXECUTE fields on panels DSNTIPRB through DSNTIPRP to specify different access lists when you run the installation CLISTS.

Before running job DSNTIRT, review the configuration statements under the CFIGIN DD statement to verify the settings of the WLMENV, GRANTTO, and optional PKGOWNER parameters.
Procedure

To install the Db2-supplied routines:

1. Run job DSNTIJRT.
2. After DSNTIJRT successfully completes, run job DSNTIJRV to validate the installation of the routines.
3. If you did not perform any of the following steps when you originally installed the Db2 subsystem, perform them for any of the following routines that you plan to begin using in Db2 12.
   - The stored procedures for XML schema support
   - The administrative task scheduler routines and administrative enablement routines
   - The CICS transaction invocation procedure
   - The common SQL API stored procedures
   - The WebSphere MQ user-defined functions
   - The web service user-defined functions

What to do next

After installing the Db2-supplied routines, you can choose to disable any of the routines that are unnecessary or unusable on your subsystem.

Related concepts:
Configuration control statements for Db2-supplied routines

Related tasks:
- Granting privileges for executing stored procedures and stored procedure packages (Managing Security)
- Disabling Db2-supplied routines for validation

Related reference:
Core WLM environments for Db2-supplied routines

Job DSNTIJRT:

Job DSNTIJRT installs and configures Db2-supplied routines by executing program DSNTRIN.

DSNTIJRT provides a configuration control statement for each Db2-supplied routine and the mode for processing those statements. The configuration control statement specifies the WLM environment, list of authorization IDs that are allowed to execute the routine, and optional package owner for the routine. DSNTRIN reads and validates those statements and uses them to create and customize routines and supporting objects.

Format

DSNTRIN accepts the following parameters:

DB2SSN
   Identifies the Db2 subsystem on which to install and configure the Db2-supplied routines.

MODE
   During SQL processing, DSNTRIN configures each routine with the WLM
environment specified in the configuration control statement for that routine. Processing behavior depends on the DSNTRIN run mode, the Db2 catalog mode, and which objects already exist.

**INSTALL**

This is the normal operating mode. Use it when installing or migrating to Db2.

Create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the routine already exists, it is updated, if necessary, to use the WLM environment specified in the configuration control statement for that routine. If the package for a routine already exists, it is not bound again unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or created global temporary table. If a supporting object is used by more than one routine, the object is created when one of those routines is first created.

**REINSTALL**

This mode behaves like INSTALL mode, except that existing Db2-supplied routines and created global temporary tables are dropped ahead of other processing. Existing related databases, table spaces, tables, and indexes are not dropped, except as discussed in the special notices section of the DSNTIJRT job prolog.

Use REINSTALL mode only if you have a specific need to drop and re-create all Db2-supplied routines. Before running the job, verify that each GRANTTO statement in DSNTIJRT specifies all users who are authorized to use that routine.

In particular, use this mode to create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the package for a routine already exists, it is not rebound unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or created global temporary table. Existing routines and any related created global temporary tables are first dropped before they are re-created. Related databases are not dropped if they already exist, but they are created if they do not exist. If a supporting object is used by more than one routine, the object is created when one of those routines is first created. Drop processing of routines and related objects that do not exist is bypassed.

**REINITIALIZE**

This mode behaves like REINSTALL mode, except that existing databases, table spaces, tables, and indexes used by the Db2-supplied routines are also dropped ahead of other processing.

REINITIALIZE mode drops and reinitializes databases such as the XSR database that may contain information needed by Db2 for other processing. Therefore, use REINITIALIZE mode only when advised to do so by IBM Support. Before running the job, verify that each GRANTTO statement in DSNTIJRT specifies all users who are authorized to use that routine.

In particular, use this mode to create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the package for a routine already exists, it is not rebound unless the DBRM has been
updated or the package has been invalidated. Create, initialize, and grant
access to any related objects required by the routine, such as a database or
created global temporary table. Existing routines, existing related created
global temporary tables, and existing related databases are first dropped
before they are re-created. After each SQL statement, a COMMIT statement
is issued. If a supporting object is used by more than one routine, the
object is created when one of those routines is first created.

**INSTALL-PREVIEW**

Simulate INSTALL mode without updating the system. Generated JCL with
embedded DDL and bind statements is outputted to the JCLOUT DD
statement.

**REINSTALL-PREVIEW**

Simulate REINSTALL mode without updating the system. Generated JCL
with embedded DDL and bind statements is outputted to the JCLOUT DD
statement.

**REINITIALIZE-PREVIEW**

Simulate REINITIALIZE mode without updating the system. Generated
JCL with embedded DDL and bind statements is outputted to the JCLOUT
DD statement.

**AUTHID**

DSNTRIN uses this value as the CURRENT SQLID for creating and
configuring most Db2-supplied routines. If DEFPKOWN is not specified,
DSNTRIN also uses this value as the default OWNER for binding packages for
the Db2-supplied routines that do not have the PKGOWNER specified in the
configuration control statement.

**SECDEFID**

DSNTRIN uses this value as the CURRENT SQLID setting when creating and
configuring Db2-supplied routines that are defined with the SECURITY
DEFINER option.

The ID that you specify for SECDEFID must be an ID that can be used as a
logon ID, because WLM uses the ID to execute Db2-supplied routines that are
defined with SECURITY DEFINER.

**DEFPKOWN**

Specifies the default OWNER for binding the package for a Db2-supplied
routine when the PKOWNER parameter of the routine’s configuration control
statement is not specified. If DEFPKOWN is not specified, DSNTRIN uses the
AUTHID value as the default package owner.

The DEFPKOWN parameter is required only when the AUTHID parameter is
set to SYSINSTL. If the INSTALL PKG OWNER field on panel DSNTIPG is
specified, the installation CLIST uses that value to set DEFPKOWN during the
editing of DSNTIJRT. Otherwise, it uses the value specified in the ROUTINES
CREATOR field on panel DSNTIPG. If SYSINSTL is specified on panel
DSNTIPG, the installation CLIST adds a notice to the DSNTIJRT prolog to
explain additional requirements.

Specify an installation system administrator ID if you use the installation
SYSOPR authority to execute this job for Db2 installation. Specify an
authorization ID that is granted the system DBADM and DATAACCESS
authorities if you use the installation SYSOPR authority to execute this job for
Db2 migration.
Input

DSNTRIN allocates the following DD statements for input:

**DBRMLIB**
Location of the DBRMs for Db2-supplied routines. This is typically the "prefix.SDSNDBRM" target library.

**SYSUT1**
Internal work file for the Db2 installation bind interface (DSNTRBI).

**CFGIN**
Location of the configuration control statements for Db2-supplied routines.

**DB2OPT**
This required DD can contain the following optional parameters:

**STOGROUP**
Specifies a storage group to use for creating databases and table spaces that are used by the routines that are supplied with Db2. If this parameter is not specified, the default SYSDEFLT storage group is used.

**INDEXSTOG**
Specifies a storage group to use for creating indexes instead of the default SYSDEFLT storage group.

**BP4K**
Specifies an alternate 4-KB buffer pool for base table spaces that are created by DSNTRIN. The default 4-KB buffer pool is BP0.

**BP8K**
Specifies an alternate 8-KB buffer pool for base table spaces that are created by DSNTRIN. The default 8-KB buffer pool is BP8K0.

**BP16K**
Specifies an alternate 16-KB buffer pool for base table spaces that are created by DSNTRIN. The default 16-KB buffer pool is BP16K0.

**BP32K**
Specifies an alternate 32-KB buffer pool for base table spaces that are created by DSNTRIN. The default 32-KB buffer pool is BP32K.

**LOBBP8K**
Specifies an alternate 8-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 8-KB buffer pool is BP8K0.

**LOBBP16K**
Specifies an alternate 16-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 16-KB buffer pool is BP16K0.

**LOBBP32K**
Specifies an alternate 32-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 32-KB buffer pool is BP32K0.

**IMS_SECURITY**
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied IMS routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 IMS routines are altered if needed to conform with the SECURITY type that is specified here.

**MQ_SECURITY**
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied MQ routines. SECURITY Db2 is assumed when this keyword is not
specified. Existing Db2 MQ routines are altered if needed to conform with the SECURITY type that is specified here.

**SOAP_SECURITY**

The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied web services/SOAP routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 SOAP routines are altered if needed to conform with the SECURITY type that is specified here.

**Note:** DSNTRIN does not create or activate this storage group or any of these buffer pools.

**Output**

DSNTRIN allocates the following DD statements for output:

**SYSPRINT**

Destination for echo of configuration control statements that are read and for messages that are generated during processing. Each processed routine generates at least one status message.

**SYSTSPRT**

Internal message file for the Db2 installation bind interface (DSNTRBI).

**CFIGOUT**

Destination for configuration control statements that are read by DSNTRIN. This output is useful for diagnostic purposes when DSNTRIN reports an error in a configuration control statement.

**SQLOUT**

Destination for DDL statements that are generated and processed by DSNTRIN. These statements are useful for diagnostic purposes.

**BINDOUT**

Destination for bind statements that are generated and processed by DSNTRIN. These statements are useful for diagnostic purposes.

**JCLOUT**

Destination for a generated sample job that is generated when one of the PREVIEW modes is specified. The job contains steps that process inline DDL and bind statements that are needed to install and configure missing, downlevel, or incorrectly configured Db2-supplied routines. This DD is required if you specify a PREVIEW mode.

Example of executing DSNTRIN in INSTALL mode:

```plaintext
//DSNTRIN  EXEC  PGM=DSNTRIN,COND=(4,LT),
//          PARM=('DB2SSN(DSN)  MODE(INSTALL)',
//            'AUTHID(ADMFO01)  SECDEFID(ADMFO02)',
//            'DEFPKOWN(SYSADM)'),
//
//           DBRMLIB  DD  DISP=SHR,DSN=DSN!!0.SDSNDBRM
//           SYSPRI DD  SYSOUT=*,DCB=(RECFM=FB,LRECL=133)
//           SYSTSPRT DD  SYSOUT=*,DCB=(RECFM=FB,LRECL=133)
//           CFIGOUT  DD  SYSOUT=*
//           SQLOUT  DD  SYSOUT=*
//           BINDOUT  DD  SYSOUT=*
//           DB2OPT  DD  *
//           STOGROUP(DB2_ROUTINES_GROUP)
//           INDEXSTOG(DB2_INDEX_GROUP)
//           BP4K(BPO)
```
Example of executing DSNTRIN in INSTALL-PREVIEW mode:

```
//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
//   PARM=('DB2SSN(DSN) MODE(INSTALL-PREVIEW)',
//      'AUTHID(ADMF001) SECDEFID(ADMF002)',
//      'DEFPKOWN(SYSADM)'),

//DBRMLIB DD DISP=SHR,DSN=DSN!0.SDSNDBRM
//SYSUT1 DD UNIT=SYSDA,SPACE=(27930,(10,5)),
//   DCB=(RECFM=FB,LRECL=133)
//SYSPRINT DD SYSOUT**,DCB=(RECFM=FB,LRECL=133)
//SYSTSPRT DD SYSOUT**
//CFGOUT DD SYSOUT**
//SOQLOUT DD SYSOUT**
//BINDOUT DD SYSOUT**
//JCLOUT DD DSN=USER.JCLLIB(DB2INST),DISP=SHR
//DB2OPT DD *
//STOGROUP(DB2_ROUTINES_GROUP)
//INDEXSTOG(DB2_INDEX_GROUP)
BP4K(BP0)
BP8K(BP8K0)
BP16K(BP16K0)
BP32K(BP32K)
LOBPP8K(BP8K0)
LOBBP16K(BP16K0)
LOBBP32K(BP32K0)
IMS_SECURITY(DB2)
MQ_SECURITY(DB2)
SOAP_SECURITY(DB2)
//CFGIN DD *
...
SYSPROC.DSNWSPM
WLMENV(DSNWLM_NUMTCB1)
GRANTTO(PUBLIC)

SYSPROC.DSNWZP
WLMENV(DSNWLM_NUMTCB1)
GRANTTO(PUBLIC)
...
//*
```

Related concepts:
- Configuration control statements for Db2-supplied routines
Installation and migration jobs that the CLIST tailors

Related reference:
Core WLM environments for Db2-supplied routines

Job DSNTIJRV:

Job DSNTIJRV validates the Db2-supplied routines by executing program
DSNTRVFY. You should validate these routines after Db2 installation and after
each phase of migration.

DSNTRVFY validates a single routine, multiple routines, or all Db2-supplied
routines. DSNTRVFY produces a report that identifies routines that passed the
validation and possible problem areas for routines that failed.

Authorizations

If the RACF facility class is active and a profile was defined for
MVSADMIN.WLM.POLICY, the user that submits this job requires READ access.

If the RACF OPERCMDS class is active and a profile was defined for
MVS.MCSOPER.*, the user ID that submits this job requires READ access.
Alternatively, you can add a discrete profile for the DSNTRVFY console that is
used in this job.

Format

DSNTRVFY accepts the following required parameters:

DB2SSN(ssid)
Identifies which Db2 subsystem to connect to for validating the Db2-supplied
routines.

ROUTINE(name)
Identifies which Db2-supplied routine needs to be validated. The following
values are supported for name:

- A routine name in the form of schema-name.routine-specific-name.
- The keyword DD:SYSIN to specify a list of routines. List the routines in the
  form of schema-name.routine-specific-name under the SYSIN DD statement with
  one routine per line.
- The keyword ALL to specify all Db2-supplied routines.

DSNTRVFY also accepts the following optional parameters:

AUTHID(authorization ID)
DSNTRVFY uses this value as the CURRENT SQLID for issued SQL
statements. Also, if PKOWNER(pkowner) is not specified, DSNTRVFY uses this
value as the OWNER for BIND PACKAGE and BIND PLAN statements. If this
value is not specified, the primary authorization ID is used for BIND
PACKAGE, BIND PLAN, and issued SQL statements.

PKOWNER(pkowner)
Specifies the default owner for binding the DSNTRVFY plan and package. If
the INSTALL PKG OWNER field on panel DSNTIPG is specified, the
installation CLIST uses that value to set PKOWNER(pkowner) during the
editing of DSNTIJRV. Otherwise, it uses the value specified in the ROUTINES
CREATOR field on panel DSNTIPG. If SYSINSTL is specified on panel DSNTIPG, the installation CLIST adds a notice to the DSNTIJRV prolog to explain additional requirements.

If you use an authorization ID with the installation SYSOPR authority to execute job DSNTIJRV, the ID must be explicitly granted the MONITOR2 system privilege. If you set AUTHID to SYSINSTL for running the SYSPROC.WLM_REFRESH stored procedure, you must also define SYSINSTL in RACF.

Output

DSNTRVFY allocates the following required DD statements for output:

**SYSPRINT**
Contains messages generated during verification processing. Each processed routine can generate one or more messages.

**SYSTSPRT**
DSNTRVFY writes bind output to SYSTSPRT if an error occurs during bind processing.

The following JCL example is for validating a single routine:

```jcl
//DSNTRYF EXEC PGM=DSNTRVFY,
   //   PARM='DB2SSN(VA1A) ROUTINE(SYSPROC.DSNUTILS)'
   //DBRMLIB DD DSN=DB2A.SDSNDBRM,DISP=SHR
   //SYSTSPRT DD SYSOUT**
   //SYSPRINT DD SYSOUT**
   //SYSIN DD *
```

The following JCL example is for validating multiple routines:

```jcl
//DSNTRYF EXEC PGM=DSNTRVFY,
   //   PARM='DB2SSN(VA1A) ROUTINE(DD:SYSIN)'
   //DBRMLIB DD DSN=DB2A.SDSNDBRM,DISP=SHR
   //SYSTSPRT DD SYSOUT**
   //SYSPRINT DD SYSOUT**
   //SYSIN DD *
   //SYSPROC.DSNWZP
   //SYSPROC.DSNACCOR
   //SYSPROC.DSNUTILS
   //SYSPROC.DSNUTILU
```

The following JCL example is for validating all routines:

```jcl
//DSNTRYF EXEC PGM=DSNTRVFY,COND=(4,LT),
   //   PARM=('DB2SSN(DSN) AUTHID(SYSINSTL)',
   //     'PKOWNER(ADMF001) ROUTINE(ALL)')
   //DBRMLIB DD DSN=DB2A.SDSNDBRM,DISP=SHR
   //SYSTSPRT DD SYSOUT**
   //SYSPRINT DD SYSOUT**
   //SYSIN DD *
```

Processing

DSNTRVFY completes the following validation:

- Verifies that the routine is defined to the SYSIBM.SYSROUTINES catalog.
• Verifies that the collection ID defined in the SYSIBM.SYSROUTINES catalog table matches the collection ID used to bind the package.
• Verifies that the routine is currently defined with the Db2-reserved WLM application environment name DSNWLM_NOT_ENABLED, indicating that it is not enabled to run on the current Db2 subsystem. If it is, then a warning message is issued and the validation is bypassed.
• Verifies if the routine can be executed on the current Db2 catalog mode.
• Verifies the APF authorization requirements.
• Verifies if the WLM application environment is defined and available for use.
• Verifies if the NUMTCB setting is valid or within the recommended range.
• Verifies if the required DD statements for some of the routines are defined.
• Verifies if the WLM environment assigned to Java routines is enabled for Java.
• Verifies security coexistence requirements (for example, some routines should run on only their own WLM application environment).
• Verifies that the selected routine can be called successfully. Certain routines such as DSNACICS, DSNAIMS, MQSeries functions, and most debugger routines are restricted to basic validation and are not called.

During the running of this job, the following message might be issued on the system console:
IEFC452I  DB2UDSMD - JOB NOT RUN - JCL ERROR

In most cases, this message is normal and can be disregarded. If you use IBM Optim Development Studio, see [Debugging stored procedures on Db2 for z/OS with Optim Development Studio, Part 2 (IBM developerWorks)] for information about DB2UDSMD.

If the validation is successful, the following message is issued:
DSNT027I  DSNTRVFY VALIDATION FOR schema.routine-specific-name SUCCESSFUL

If the validation fails, messages are issued to explain the reasons for failure. Examine these messages and make necessary adjustments before running the validation job again.

Input

DSNTRVFY requires the following DD statements for input:

**DBRMLIB**
- Identifies the library where the DBRM for the DSNTRVFY program is located.
- This is typically the *prefix.SDSNDBRM* target library.

**SYSIN**
- Contains the list of Db2 routines. This is required when the validation type is for multiple routines (DD:SYSIN).

**SYSUT1**
- Work data set required for bind processing.

Limitations

The following limitations apply to the DSNTRVFY program:
• The name of the routine in the form *schema-name.routine-specific-name* is limited to 72 characters, the maximum line length in JCL.
• Only Db2 12 Db2-supplied routines are supported for validation.
• Some routines, if enabled, receive basic validation to ensure that they are
defined correctly and assigned to a valid WLM application environment.
However, validation will not extend to calling these routines for one or more of
the following reasons:
  – Optional system support such as CICS, IMS, or MQSeries is required.
  – External setup on a remote server or workstation is required.
  – A call could introduce or modify data unexpectedly in a production
  environment.

The Db2-supplied routines that are restricted to basic validation are:

– DB2DEBUG.CREATE_SESSION
– DB2DEBUG.DESTROY_SESSION
– DB2DEBUG.GET_REPORT
– DB2DEBUG.LIST_SESSION
– DB2DEBUG.PUT_COMMAND
– SYSFUN.SYSTS_ENCRYPT
– SYSFUN.XSLTRANSFORM
– SYSPROC.ADMIN_COMMAND_UNIX
– SYSPROC.ADMIN_JOB_CANCEL
– SYSPROC.ADMIN_JOB_FETCH
– SYSPROC.ADMIN_JOB_QUERY
– SYSPROC.ADMIN_JOB_SUBMIT
– SYSPROC.DBG_ENDSESSIONMANAGER
– SYSPROC.DBG_INITIALIZECLIENT
– SYSPROC.DBG_RECVCLIENTREPORTS
– SYSPROC.DBG_SENDCLIENTREQUESTS
– SYSPROC.DBG_SENDCLIENTCOMMANDS
– SYSPROC.DBG_TERMINATECLIENT
– SYSPROC.DSNACICS
– SYSPROC.DSNAIMS
– SYSPROC.DSNAIMS2
– SYSPROC.DSNLEUSR
– SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY
– SYSPROC.SYSTS_ALTER
– SYSPROC.SYSTS_CREATE
– SYSPROC.SYSTS_DROP
– SYSPROC.SYSTS_RESTORE
– SYSPROC.SYSTS_START
– SYSPROC.SYSTS_STOP
– SYSPROC.SYSTS_neurons
– SYSPROC.SYSTS_UPDATE
– All MQSeries, web services, and SOAP functions (those with schema DB2MQ,
  DB2MQ1C, DB2MQ2C, or DB2XML)
Notes

Temporary stored procedures used by DSNTIJRV

DSNTRVfy creates, calls, and drops two temporary stored procedures. These temporary stored procedures are SYSPROC.DSNWLM_INSTALL_INFOJ and SYSPROC.DSNWLM_INSTALL_INFO. Both of these stored procedures exist only when DSNTRVfy is executing. DSNTRVfy calls SYSPROC.DSNWLM_INSTALL_INFOJ to validate the WLM environment that is assigned to each Db2-supplied Java routine. DSNTRVfy calls SYSPROC.DSNWLM_INSTALL_INFO to validate the WLM environment that is assigned to each Db2-supplied non-Java routine that runs in a WLM environment.

Using DSNTIJRV to verify 64-bit Java routines

Java on z/OS has two types of architectures: a 31-bit version and a 64-bit version. If your Java routines run in a 31-bit Java virtual machine (JVM), the startup procedure for the WLM environment in which the routines run must initialize a 31-bit environment. If you are running Java routines in a 64-bit JVM, the startup procedure for the WLM environment in which your Java routines run must initialize a 64-bit environment. The program that initializes a 31-bit WLM environment is DSNX9WLM. The program that initializes a 64-bit WLM environment is DSNX9WJM.

For example, you might use the Db2 core Java WLM environment, DSNWLM_JAVA, for your Java routines. By default, the DSNWLMJ startup procedure for DSNWLM_JAVA runs program DSNX9WLM. If your JAVA_HOME environment variable specifies the path to a 31-bit JVM, such as /usr/lpp/java/IBM/J8.0, you need to run program DSNX9WLM to initialize the WLM environment. However, if your JAVA_HOME variable specifies a path to a 64-bit JVM, such as /usr/lpp/java/8.0_64, you need to run program DSNX9WJM to initialize the WLM environment. If you switch to a 64-bit JVM for your Java stored procedures, but do not update the initialization program to DSNX9WJM, DSNTIJRV fails.

Related concepts:

- WLM address space startup procedure for Java routines (Db2 Application Programming for Java)

Migration step 25: Convert EXPLAIN tables to the current format

When you migrate to Db2 12, you should convert your EXPLAIN tables to the current (Db2 12) format.

A table that is not in the current format has fewer columns than are available in the current release, or it has columns that have a different length or data type than expected in the current release.

About this task

In Db2 12, statements or commands that invoke EXPLAIN processing return SQLCODE +20520 reason code 2 if an EXPLAIN table is in Db2 11 format. Db2 returns SQLCODE -2008 reason code 2 for statements or commands that invoke EXPLAIN processing if the EXPLAIN tables are in a pre-Db2 10 format. As of Db2 11, EXPLAIN tables are required to be encoded in Unicode. If an EXPLAIN table is encoded in EBCDIC, SQLCODE -878 is returned.
Procedure

To convert EXPLAIN tables:

1. You can identify EXPLAIN tables that are in a pre-Db2 12 format by running the following query:

```sql
WITH OLD_FORMAT_EXPLAIN_TABLES (FULL_NAME) AS
(SELECT STRIP(TBCREATOR) || '.' || STRIP(TBNAME)
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME IN ('DSN_COLDIST_TABLE',
                 'DSN_DET_COST_TABLE',
                 'DSN_FUNCTION_TABLE',
                 'DSN_KEYTGDIST_TABLE',
                 'DSN_PGRANGE_TABLE',
                 'DSN_PGROUP_TABLE',
                 'DSN_PREDICAT_TABLE',
                 'DSN_PREDICATE_SELECTIVITY',
                 'DSN_PROFILE_HISTORY',
                 'DSN_PROFILE_TABLE',
                 'DSN_PTASK_TABLE',
                 'DSN_QUERY_TABLE',
                 'DSN_QUERYINFO_TABLE',
                 'DSN_SORT_TABLE',
                 'DSN_SORTKEY_TABLE',
                 'DSN_STAT_FEEDBACK',
                 'DSN_STATEMENT_CACHE_TABLE',
                 'DSN_STATEMENT_TABLE',
                 'DSN_STRUCT_TABLE',
                 'DSN_VIEWREF_TABLE',
                 'DSN_VIRTUAL_INDEXES',
                 'DSN_VIRTUAL_KEYTARGETS',
                 'PLAN_TABLE')
GROUP BY TBCREATOR, TBNAME
HAVING (TBNAME = 'DSN_COLDIST_TABLE'
     AND MAX(COLNO) < 22)
 OR (TBNAME = 'DSN_DET_COST_TABLE'
     AND MAX(COLNO) < 138)
 OR (TBNAME = 'DSN_FUNCTION_TABLE'
     AND MAX(COLNO) < 21)
 OR (TBNAME = 'DSN_KEYTGDIST_TABLE'
     AND MAX(COLNO) < 22)
 OR (TBNAME = 'DSN_PGRANGE_TABLE'
     AND MAX(COLNO) < 16)
 OR (TBNAME = 'DSN_PGROUP_TABLE'
     AND MAX(COLNO) < 43)
 OR (TBNAME = 'DSN_PREDICAT_TABLE'
     AND MAX(COLNO) < 43)
 OR (TBNAME = 'DSN_PREDICATE_SELECTIVITY'
     AND MAX(COLNO) < 17)
 OR (TBNAME = 'DSN_PROFILE_HISTORY'
     AND MAX(COLNO) < 15)
 OR (TBNAME = 'DSN_PROFILE_TABLE'
     AND MAX(COLNO) < 15)
 OR (TBNAME = 'DSN_PTASK_TABLE'
     AND MAX(COLNO) < 26)
 OR (TBNAME = 'DSN_QUERY_TABLE'
     AND MAX(COLNO) < 16)
 OR (TBNAME = 'DSN_QUERYINFO_TABLE'
     AND MAX(COLNO) < 18)
)
OR (TBNAME = 'DSN_SORT_TABLE'
AND MAX(COLNO) < 17)
OR (TBNAME = 'DSN_SORTKEY_TABLE'
AND MAX(COLNO) < 22)
OR (TBNAME = 'DSN_STATEMENT_CACHE_TABLE'
AND MAX(COLNO) < 93)
OR (TBNAME = 'DSN_STATEMENT_TABLE'
AND MAX(COLNO) < 24)
OR (TBNAME = 'DSN_STRUCT_TABLE'
AND MAX(COLNO) < 21)
OR (TBNAME = 'DSN_VIEWREF_TABLE'
AND MAX(COLNO) < 13)
OR (TBNAME = 'DSN_VIRTUAL_INDEXES'
AND MAX(COLNO) < 150)
OR (TBNAME = 'DSN_VIRTUAL_INDEXES'
AND MAX(COLNO) < 16)
OR (TBNAME = 'PLAN_TABLE'
AND MAX(COLNO) < 67)
UNION
SELECT STRIP(TBCREATOR) || '.' || STRIP(TBNAME)
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME IN ( 'DSN_DECOST_TABLE'
, 'DSN_FUNCTION_TABLE'
, 'DSN_PGRANGE_TABLE'
, 'DSN_PGROUP_TABLE'
, 'DSN_PGROUP_TABLE'
, 'DSN_PROFILE_HISTORY'
, 'DSN_PROFILE_TABLE'
, 'DSN_PTASK_TABLE'
, 'DSN_QUERY_TABLE'
, 'DSN_SORT_TABLE'
, 'DSN_SORTKEY_TABLE'
, 'DSN_STATEMENT_CACHE_TABLE'
, 'DSN_STATEMENT_RUNTIME_INFO'
, 'DSN_STATEMENT_TABLE'
, 'DSN_STRUCT_TABLE'
, 'DSN_VIEWREF_TABLE'
, 'DSN_VIRTUAL_INDEXES'
, 'PLAN_TABLE'
)
AND ( (NAME = 'ACCESSCREATOR' AND LENGTH <= 128)
OR (NAME = 'ACCESSNAME' AND LENGTH <= 128)
OR (NAME = 'APPLNAME' AND LENGTH <= 24)
OR (NAME = 'COLLID' AND LENGTH <= 128)
OR (NAME = 'CORRELATION_NAME' AND LENGTH <= 128)
OR (NAME = 'CREATOR' AND LENGTH <= 128)
OR (NAME = 'FUNCTION_NAME' AND LENGTH <= 128)
OR (NAME = 'FUNCTION_TEXT' AND LENGTH <= 1500)
OR (NAME = 'GROUP_MEMBER' AND LENGTH <= 24)
OR (NAME = 'HINTUSED' AND LENGTH <= 128)
OR (NAME = 'OPHINT' AND LENGTH <= 128)
OR (NAME = 'PATH' AND LENGTH <= 2048)
OR (NAME = 'PLANNAME' AND LENGTH <= 24)
OR (NAME = 'PROGNAME' AND LENGTH <= 128)
OR (NAME = 'REMARKS' AND LENGTH <= 762)
OR (NAME = 'SCHEMA_NAME' AND LENGTH <= 128)
OR (NAME = 'SPEC_FUNC_NAME' AND LENGTH <= 128)
OR (NAME = 'TNAMESPACE' AND LENGTH <= 128)
OR (NAME = 'VERSION' AND LENGTH <= 128)
OR (NAME = 'VIEW_CREATOR' AND LENGTH <= 128)
OR (NAME = 'VIEW_NAME' AND LENGTH <= 128)
)
)
SELECT DISTINCT(FULL_NAME)
FROM OLD_FORMAT_EXPLAIN_TABLES
ORDER BY FULL_NAME;
2. If the query identifies any EXPLAIN tables that are not in Db2 12 format, customize and run job DSNTIJXA.
   Job DSNTIJXA calls REXX exec DSNTXTA, which alters to the Db2 12 format all EXPLAIN tables or EXPLAIN tables that belong to the specified creator ID.
   To customize the job, provide the creator ID of the tables to migrate to the Db2 12 format, a Db2 subsystem name, and an authorization ID. To migrate all EXPLAIN tables, specify an asterisk as the creator ID.

   Related reference:
   - [EXPLAIN tables (Db2 Performance)](#)

### Migration step 26: Verify views: DSNTIJVR
During migration to Db2 12, some views might be marked with view regeneration errors. If any views are marked with view regeneration errors, you can regenerate those views again.

#### Procedure

To identify views that are marked with regeneration errors, and generate the ALTER VIEW REGENERATE statements to correct them:

1. Run job DSNTIJVR.
2. Use dynamic SQL processing tools, such as SPUFI or DSNTEP2, to process any statements that are generated by DSNTIJVR.

### Migration step 27: Take an image copy of the Db2 12 catalog: DSNTIJIC
For backup purposes, create a copy of the Db2 12 catalog and directory.

#### Procedure

To create an image copy of the directory and catalog:

Run the Db2 12 DSNTIJIC job.

### Migration step 28: Verify migration of Db2 11 IVP objects to Db2 12 (optional)
Verify the migration of IVP objects to Db2 12 by running some of the sample jobs from your previous release.

#### About this task

If all of the local Db2 objects from Db2 11 still exist (that is, if you have not run job DSNTEJ0), follow this procedure.
Procedure

To verify the migration of IVP objects in your Db2 12 system:

1. Change the JOBLIB statements to point to prefix.SDSNLOAD.
2. Ensure that the DSN8EAE1 module that you created when you originally ran the Db2 11 sample jobs is copied to prefix.SDSNEXIT. DSN8EAE1 is an EDITPROC that is used by the employee sample table.
3. Edit the Db2 11 sample jobs before running them. Do not run all the sample jobs. Run only the specific jobs and job steps that are listed in the following steps.
4. Test the migration of the IVP phase 2 applications from your previous version:
   a. DSNTEJ2A: Perform all except the first two steps of job DSNTEJ2A. Expect a return code of 4 because table spaces DSN8D11U.NEWDEPT and DSN8D11U.NEWPHONE are placed in COPY-pending states.
   b. DSNTEJ2C: Issue only the RUN PROGRAM(DSN8BC3) PLAN(DSN8BH11) statement in step PH02CS04.
   c. DSNTEJ2D: Issue only the RUN PROGRAM(DSN8BD3) PLAN(DSN8BD11) statement in step PH02DS03.
   d. DSNTEJ2E: Issue only the RUN PROGRAM(DSN8BE3) PLAN(DSN8BE11) statement in step PH02ES04.
   e. DSNTEJ2F: Issue only the RUN PROGRAM(DSN8BF3) PLAN(DSN8BF11) statement in step PH02FS03.
   f. DSNTEJ2P: Run only step PH02PS05.
5. Test the migration of the IVP phase 3 applications from your previous version:
   a. Do not run job DSNTEJ3C or DSNTEJ3P.
   b. If you want to test the Db2 11 ISPF-CAF applications under Db2 12, place the Db2 11 SDSNSPFP panel library ahead of the Db2 12 SDSNSPFP panel library in the ISPPLIB concatenation. This placement is necessary so that the plans that are migrated from Db2 11 can be used. Remove the Db2 11 SDSNSPFP library from your ISPPLIB concatenation when you are finished testing the Db2 11 IVP applications under Db2 12.

What to do next

Do not run any other Db2 11 sample jobs.

Related concepts:
- Dynamic SQL and the ISPF/CAF application

Related tasks:
- Ensure that sample objects from your previous release are available

Migration step 29: Activate function level 500 or higher

The use of most new capabilities and enhancements that Db2 12 introduces is prevented until you activate function level 500 or higher. You cannot fallback to Db2 11 and Db2 12 cannot coexist with Db2 11 after you activate function level 500.

Before you begin

Important: In data sharing, ensure that the data sharing group has no active Db2 11 members. For instructions for migrating data sharing members to Db2 12, see Migrating subsequent members of a group to Db2 12.
**Important:** Do not issue the ACTIVATE command or run job DSNTIJAF for activation of function level 500 or higher until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

**Important:** Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

Determine the function level to activate. In most cases, you can activate a higher function level without separately activating each lower function level above the currently activated function level. However, activating a higher function level also results in the activation of all lower function levels. Before activating a function level, familiarize yourself with the new capabilities and changes that all lower function levels introduce:

- [Incompatible changes summary for function levels 501 and higher (Db2 for z/OS What's New?)](https://www.ibm.com/support/knowledgecenter/STAXYR_12.1.0/com.ibm.db2.luw.sql.ref.doc/doc/rzsrcf91.html)

**About this task**

To activate Db2 12 new function, you activate function level 500 or higher. Successful activation of function level 500 or higher disables coexistence with and fallback to Db2 11. In function level 100, fallback and coexistence with Db2 11 remain possible.

**[FL 502]** Before you can tailor the Db2 catalog for function level 502, function level 500 or 501 must be activated first. However, activating a lower star (*) function level does not prevent the catalog from being tailored for function level 502.

**Overview of Db2 12 new function availability**

The availability of new function depends on the type of enhancement, the activated function level, and the application compatibility levels of applications. In the initial Db2 12 release, most new capabilities are enabled only after the activation of function level 500 or higher.

**Virtual storage enhancements**

Virtual storage enhancements become available at the activation of the function level that introduces them or higher. Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release. That is, activation of function level 500 introduces no virtual storage enhancements.

**Subsystem parameters**

New subsystem parameter settings are in effect only when the function level that introduced them or a higher function level is activated. Most subsystem parameter changes in the initial Db2 12 release take effect in function level 500. For a list of these changes, see [Subsystem parameter changes in Db2 12 (Db2 for z/OS What's New?)](https://www.ibm.com/support/knowledgecenter/STAXYR_12.1.0/com.ibm.db2.luw.sql.ref.doc/doc/rztlnp70.html).
Optimization enhancements

Optimization enhancements become available after the activation of the function level that introduces them or higher, and full prepare of the SQL statements. When a full prepare occurs depends on the statement type:

- For static SQL statements, after bind or rebind of the package
- For non-stabilized dynamic SQL statements, immediately, unless the statement is in the dynamic statement cache
- For stabilized dynamic SQL statements, after invalidation, free, or changed application compatibility level

Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. That is, function level 500 introduces no optimization enhancements.

SQL capabilities

New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous Db2 releases, including Db2 11 and Db2 10. For details, see Application compatibility levels in Db2 (Db2 Application programming and SQL).

Procedure

To activate Db2 12 new function for the first time, complete the following steps:

1. Issue a DISPLAY GROUP command to check that the code level of the Db2 subsystem or each data sharing group member supports your target function level. In the DSN7100I message, the DB2_LVL column indicates the code level. For more information and examples, see Determining the Db2 code level, catalog level, and function level (Db2 for z/OS What's New?).

2. If necessary, apply maintenance, such as PTFs and RSUs, to bring your Db2 subsystem or data sharing group members up to the required code level for your target function level.

3. Run the installation CLIST, as described in Tailoring Db2 12 installation and migration jobs with the CLIST.
   a. On panel DSNTIPA1, specify ACTIVATE in the INSTALL TYPE field.
   b. On panel DSNTIP00, specify V12R1M500 or higher in the TARGET FUNCTION LEVEL field.
   c. Optional: If all Db2 applications can be bound and run at the target function level, modify the APPLCOMPAT and SQLLEVEL subsystem parameter settings. Otherwise, leave the APPLCOMPAT and SQLLEVEL fields unchanged.
   d. Exit the CLIST. The output data set contains tailored jobs for the activation process.

4. If the target function level is V12R1M502 or higher, run the following jobs:
   a. Run the DSNTIJA1 job to issue an ACTIVATE command to activate function level 501. Activation of function level 500 or 501 is required before you can tailor the Db2 catalog for function level 502 or higher.
b. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

c. Run the DSNTIJTC job to run the CATMAINT utility to tailor the Db2 catalog for the target function level.

**Important:** Do not attempt to start Db2 at a lower code level after any part of the CATMAINT job for a higher function level completes. Run the CATMAINT job only after you are satisfied that Db2 can continue to run at the required code level. The code to tolerate catalog changes is contained in the code level that delivers the CATMAINT job.

5. Optional: Test the activation of the target function level, as described in [Testing Db2 function level activation (Db2 for z/OS What’s New?)](#).

6. Run the DSNTIJAF job to issue an ACTIVATE command for the target function level. If the command completes successfully, Db2 issues message DSN9022I. However, if any active member of the data sharing group is running on a previous release, the command fails, and Db2 issues message DSN9039I. Message DSNU757I indicates the result of the activate command. For example, the following message indicates successful activation of function level 500:

```
DSNU757I -DB2A DSNUGCCA *** BEGIN ACTIVATE FUNCTION LEVEL (V12R1M500)
FUNCTION LEVEL (V12R1M500) SUCCESSFULLY ACTIVATED
CATALOG LEVEL(V12R1M500)
CURRENT FUNCTION LEVEL(V12R1M500)
PREVIOUS HIGHEST FUNCTION LEVEL (V12R1M500)
HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
DSN9022I -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
```

More actions are required after the ACTIVATE command completes successfully before most types of new capabilities and enhancements in the function level can be used. For example, new SQL capabilities require that applications use the appropriate application compatibility level, and optimization enhancements apply only after full prepare of the SQL statements.

**Tip:** Proceed with the following steps only if all Db2 applications can be bound and run at the target function level. For more information, see [Controlling Db2 application compatibility (Db2 for z/OS What’s New?)](#).

7. Optional: Run the DSNTIJUZ job to modify the subsystem parameter module with the APPLCOMPAT value that was specified on panel DSNTIP00.

8. Optional: Run the DSNTIJ0Z job to run a SET SYSPARM command to bring the APPLCOMPAT subsystem parameter change online.

9. Optional: Run the DSNTIJUA job to modify the Db2 data-only application defaults module with the SQLLEVEL value that was specified on panel DSNTIP00.

**What to do next**

Complete any of the following actions:

- If the new function level includes optimization enhancements, Db2 must process a full prepare before any SQL statements can benefit. When a full prepare occurs depends on the statement type:
  - For static SQL statements, after bind or rebind of the package
  - For non-stabilized dynamic SQL statements, immediately, unless the statement is in the dynamic statement cache
  - For stabilized dynamic SQL statements, after invalidation, free, or changed application compatibility level
If you did not run the jobs to update the APPLCOMPAT and SQLLEVEL subsystem parameters, resolve any application incompatibilities and increase the application compatibility level of your applications after you are satisfied that Db2 is stable at the target function level, as described in Controlling Db2 application compatibility (Db2 for z/OS What's New?).

If you encounter regressions or other problems when you activate a Db2 12 function level, minimize the impact to your applications while you resolve the problems by following the general approaches described in Responding to problems after function level activation (Db2 for z/OS What's New?).

Related reference:

- What's new in Db2 12 function levels (Db2 for z/OS What's New?)
- Function level 500 (activated at Db2 12 installation or after migration—October, 2016) (Db2 for z/OS What's New?)
- ACTIVATE (Db2) (Db2 Commands)
- CATMAINT (Db2 Utilities)

Related information:

- DSNU757I (Db2 Messages)
- DSN9022I (Db2 Messages)
- DSN9039I (Db2 Messages)
- Video: Db2 for z/OS—Delivering New Capabilities Faster (YouTube: 1:33:25)

Migration step 30: Install and configure Db2-supplied routines that use new function: DSNTIJRT

You must run the DSNTIJRT job again after you activate a higher function level in Db2 12 to install and configure Db2-supplied routines that are dependent on that function level.

Before you begin

Activate function level 500 or higher.

About this task

Job DSNTIJRT executes DSNTRIN, which installs and customizes all Db2-supplied routines and supporting objects that were deferred until new function is activated. This job was customized when the CLIST was run in MIGRATE mode.

Procedure

To install and configure routines that are dependent on new function:

Run job DSNTIJRT

What to do next

Migration step 31: Validate Db2-supplied routines: DSNTIJRV

Related tasks:

Installing Db2-supplied routines during migration

Related reference:
Job DSNTIJRT

Job DSNTIJRT installs and configures Db2-supplied routines by executing program DSNTRIN.

DSNTIJRT provides a configuration control statement for each Db2-supplied routine and the mode for processing those statements. The configuration control statement specifies the WLM environment, list of authorization IDs that are allowed to execute the routine, and optional package owner for the routine. DSNTRIN reads and validates those statements and uses them to create and customize routines and supporting objects.

Format

DSNTRIN accepts the following parameters:

DB2SSN
Identifies the Db2 subsystem on which to install and configure the Db2-supplied routines.

MODE
During SQL processing, DSNTRIN configures each routine with the WLM environment specified in the configuration control statement for that routine. Processing behavior depends on the DSNTRIN run mode, the Db2 catalog mode, and which objects already exist.

INSTALL
This is the normal operating mode. Use it when installing or migrating to Db2.

Create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the routine already exists, it is updated, if necessary, to use the WLM environment specified in the configuration control statement for that routine. If the package for a routine already exists, it is not bound again unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or created global temporary table. If a supporting object is used by more than one routine, the object is created when one of those routines is first created.

REINSTALL
This mode behaves like INSTALL mode, except that existing Db2-supplied routines and created global temporary tables are dropped ahead of other processing. Existing related databases, table spaces, tables, and indexes are not dropped, except as discussed in the special notices section of the DSNTIJRT job prolog.

Use REINSTALL mode only if you have a specific need to drop and re-create all Db2-supplied routines. Before running the job, verify that each GRANTTO statement in DSNTIJRT specifies all users who are authorized to use that routine.

In particular, use this mode to create, bind, and grant access to the Db2-supplied routine according to the WLMENV and GRANTTO parameters of the configuration control statement. If the package for a routine already exists, it is not rebound unless the DBRM has been updated or the package has been invalidated. Create, initialize, and grant access to any related objects required by the routine, such as a database or
created global temporary table. Existing routines and any related created
global temporary tables are first dropped before they are re-created.
Related databases are not dropped if they already exist, but they are
created if they do not exist. If a supporting object is used by more than one
routine, the object is created when one of those routines is first created.
Drop processing of routines and related objects that do not exist is
bypassed.

**REINITIALIZE**

This mode behaves like REINSTALL mode, except that existing databases,
table spaces, tables, and indexes used by the Db2-supplied routines are
also dropped ahead of other processing.

REINITIALIZE mode drops and reinitializes databases such as the XSR
database that may contain information needed by Db2 for other processing.
Therefore, use REINITIALIZE mode only when advised to do so by IBM
Support. Before running the job, verify that each GRANTTO statement in
DSNTRJRT specifies all users who are authorized to use that routine.

In particular, use this mode to create, bind, and grant access to the
Db2-supplied routine according to the WLMENV and GRANTTO
parameters of the configuration control statement. If the package for a
routine already exists, it is not rebound unless the DBRM has been
updated or the package has been invalidated. Create, initialize, and grant
access to any related objects required by the routine, such as a database or
created global temporary table. Existing routines, existing related created
global temporary tables, and existing related databases are first dropped
before they are re-created. After each SQL statement, a COMMIT statement
is issued. If a supporting object is used by more than one routine, the
object is created when one of those routines is first created.

**INSTALL-PREVIEW**

Simulate INSTALL mode without updating the system. Generated JCL with
embedded DDL and bind statements is outputted to the JCLOUT DD
statement.

**REINSTALL-PREVIEW**

Simulate REINSTALL mode without updating the system. Generated JCL
with embedded DDL and bind statements is outputted to the JCLOUT DD
statement.

**REINITIALIZE-PREVIEW**

Simulate REINITIALIZE mode without updating the system. Generated
JCL with embedded DDL and bind statements is outputted to the JCLOUT
DD statement.

**AUTHID**

DSNTRIN uses this value as the CURRENT SQLID for creating and
configuring most Db2-supplied routines. If DEFPKOWN is not specified,
DSNTRIN also uses this value as the default OWNER for binding packages for
the Db2-supplied routines that do not have the PKGOWNER specified in the
configuration control statement.

**SECDEFFID**

DSNTRIN uses this value as the CURRENT SQLID setting when creating and
configuring Db2-supplied routines that are defined with the SECURITY
DEFINER option.
The ID that you specify for SECDEFID must be an ID that can be used as a logon ID, because WLM uses the ID to execute Db2-supplied routines that are defined with SECURITY DEFINER.

**DEFPKOWN**

Specifies the default OWNER for binding the package for a Db2-supplied routine when the PKOWNER parameter of the routine's configuration control statement is not specified. If DEFPKOWN is not specified, DSNTRIN uses the AUTHID value as the default package owner.

The DEFPKOWN parameter is required only when the AUTHID parameter is set to SYSINSTL. If the INSTALL PKG OWNER field on panel DSNTIPG is specified, the installation CLIST uses that value to set DEFPKOWN during the editing of DSNTIJRT. Otherwise, it uses the value specified in the ROUTINES CREATOR field on panel DSNTIPG. If SYSINSTL is specified on panel DSNTIPG, the installation CLIST adds a notice to the DSNTIJRT prolog to explain additional requirements.

Specify an installation system administrator ID if you use the installation SYSOPR authority to execute this job for Db2 installation. Specify an authorization ID that is granted the system DBADM and DATAACCESS authorities if you use the installation SYSOPR authority to execute this job for Db2 migration.

**Input**

DSNTRIN allocates the following DD statements for input:

**DBRMLIB**

Location of the DBRMs for Db2-supplied routines. This is typically the prefix.SDSNDBRM target library.

**SYSUT1**

Internal work file for the Db2 installation bind interface (DSNTRBI).

**CFIGIN**

Location of the configuration control statements for Db2-supplied routines.

**DB2OPT**

This required DD can contain the following optional parameters:

**STOGROUP**

Specifies a storage group to use for creating databases and table spaces that are used by the routines that are supplied with Db2. If this parameter is not specified, the default SYSDEFLT storage group is used.

**INDEXSTOG**

Specifies a storage group to use for creating indexes instead of the default SYSDEFLT storage group.

**BP4K**

Specifies an alternate 4-KB buffer pool for base table spaces that are created by DSNTRIN. The default 4-KB buffer pool is BP0.

**BP8K**

Specifies an alternate 8-KB buffer pool for base table spaces that are created by DSNTRIN. The default 8-KB buffer pool is BP8K0.

**BP16K**

Specifies an alternate 16-KB buffer pool for base table spaces that are created by DSNTRIN. The default 16-KB buffer pool is BP16K0.
BP32K
Specifies an alternate 32-KB buffer pool for base table spaces that are created by DSNTRIN. The default 32-KB buffer pool is BP32K.

LOBBP8K
Specifies an alternate 8-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 8-KB buffer pool is BP8K0.

LOBBP16K
Specifies an alternate 16-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 16-KB buffer pool is BP16K0.

LOBBP32K
Specifies an alternate 32-KB buffer pool for LOB table spaces that are created by DSNTRIN. The default 32-KB buffer pool is BP32K0.

IMS_SECURITY
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied IMS routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 IMS routines are altered if needed to conform with the SECURITY type that is specified here.

MQ_SECURITY
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied MQ routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 MQ routines are altered if needed to conform with the SECURITY type that is specified here.

SOAP_SECURITY
The SECURITY type (Db2, DEFINER, or USER) for the Db2-supplied web services/SOAP routines. SECURITY Db2 is assumed when this keyword is not specified. Existing Db2 SOAP routines are altered if needed to conform with the SECURITY type that is specified here.

Note: DSNTRIN does not create or activate this storage group or any of these buffer pools.

Output

DSNTRIN allocates the following DD statements for output:

SYSPRINT
Destination for echo of configuration control statements that are read and for messages that are generated during processing. Each processed routine generates at least one status message.

SYSTSPRT
Internal message file for the Db2 installation bind interface (DSNTRBI).

CFIGOUT
Destination for configuration control statements that are read by DSNTRIN. This output is useful for diagnostic purposes when DSNTRIN reports an error in a configuration control statement.

SQLOUT
Destination for DDL statements that are generated and processed by DSNTRIN. These statements are useful for diagnostic purposes.

BINDOUT
Destination for bind statements that are generated and processed by DSNTRIN. These statements are useful for diagnostic purposes.
JCLOUT

Destination for a generated sample job that is generated when one of the
PREVIEW modes is specified. The job contains steps that process inline DDL
and bind statements that are needed to install and configure missing,
downlevel, or incorrectly configured Db2-supplied routines. This DD is
required if you specify a PREVIEW mode.

Example of executing DSNTRIN in INSTALL mode:

//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
   //       PARM=('DB2SSN(DSN) MODE(INSTALL)',
   //       'AUTHID(ADMF001) SECDEFID(ADMF002)',
   //       'DEFPKOWN(SYSADM)')

//DBRMLIB DD DISP=SHR,DSN=DSN!!0.SDSNDBRM
//SYSUT1 DD UNIT=sysda,SPACE=(27930,(10,5)),
   //       DCB=(RECFM=FB,LRECL=133)
//SYSPRINT DD SYSOUT**,DCB=(RECFM=FB,LRECL=133)
//SYSTSPrT DD SYSOUT**
//CFIGOUT DD SYSOUT**
//SQLOUT DD SYSOUT**
//BINDOUT DD SYSOUT**
//DB2OPT DD *
STOGROUP(DB2_Routines_GROUP)
INDEXSTOG(DB2_INDEX_GROUP)
BP4K(BP0)
BP8K(BP8K0)
BP16K(BP16K0)
BP32K(BP32K)
LOBBP8K(BP8K0)
LOBBP16K(BP16K0)
LOBBP32K(BP32K0)
IMS_SECURITY(DB2)
MQ_SECURITY(DB2)
SOAP_SECURITY(DB2)

/*
   //CFIGIN DD *
   ...
   SYSPROC.DSNWSPM
   WLMENV(DSNWLM_NUMTCB1)
   GRANTTO(PUBLIC)
   SYSPROC.DSNWZP
   WLMENV(DSNWLM_NUMTCB1)
   GRANTTO(PUBLIC)
   ...
   */

Example of executing DSNTRIN in INSTALL-PREVIEW mode:

//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
   //       PARM=('DB2SSN(DSN) MODE(INSTALL-PREVIEW)',
   //       'AUTHID(ADMF001) SECDEFID(ADMF002)',
   //       'DEFPKOWN(SYSADM)')

//DBRMLIB DD DISP=SHR,DSN=DSN!!0.SDSNDBRM
//SYSUT1 DD UNIT=sysda,SPACE=(27930,(10,5)),
   //       DCB=(RECFM=FB,LRECL=133)
//SYSPRINT DD SYSOUT**,DCB=(RECFM=FB,LRECL=133)
//SYSTSPrT DD SYSOUT**
//CFIGOUT DD SYSOUT**
//SQLOUT DD SYSOUT**
//BINDOUT DD SYSOUT**
//JCLOUT DD DSN=USER.JCLLIB(DB2INST),DISP=SHR
//DB2OPT DD *
Migration step 31: Validate Db2-supplied routines: DSNTIJRVA

You can validate Db2-supplied routines and augment existing installation verification procedures (IVPs) for specific Db2-supplied routines.

Before you begin

- **Activate function level 500 or higher.**
- **Run job DSNTIJRVT again after activating Db2 12 new function.**

About this task

Job DSNTIJRVA validates all Db2-supplied routines and augments existing installation verification procedures (IVPs) for specific Db2-supplied routines. It also provides a report that identifies successful validations, warnings, and failures. This job was customized when the CLIST was run in MIGRATE mode.

Procedure

To validate all Db2-supplied routines and augment existing IVPs for specific Db2-supplied routines:

Run job DSNTIJRVA.

Related tasks:
- Installing Db2-supplied routines during migration

Related reference:
- Installation information for procedures and functions that are supplied with Db2
Job DSNTIJRV

Job DSNTIJRV validates the Db2-supplied routines by executing program DSNTLRVFY. You should validate these routines after Db2 installation and after each phase of migration.

DSNTRVFY validates a single routine, multiple routines, or all Db2-supplied routines. DSNTRVFY produces a report that identifies routines that passed the validation and possible problem areas for routines that failed.

Authorizations

If the RACF facility class is active and a profile was defined for MVSADMIN.WLM.POLICY, the user that submits this job requires READ access.

If the RACF OPERCMDS class is active and a profile was defined for MVS.MCSOPER.*, the user ID that submits this job requires READ access. Alternatively, you can add a discrete profile for the DSNTRVFY console that is used in this job.

Format

DSNTRVFY accepts the following required parameters:

DB2SSN(ssid)

Identifies which Db2 subsystem to connect to for validating the Db2-supplied routines.

ROUTINE(name)

Identifies which Db2-supplied routine needs to be validated. The following values are supported for name:

- A routine name in the form of schema-name.routine-specific-name.
- The keyword DD:SYSIN to specify a list of routines. List the routines in the form of schema-name.routine-specific-name under the SYSIN DD statement with one routine per line.
- The keyword ALL to specify all Db2-supplied routines.

DSNTRVFY also accepts the following optional parameters:

AUTHID(authorization ID)

DSNTRVFY uses this value as the CURRENT SQLID for issued SQL statements. Also, if PKOWNER(pkowner) is not specified, DSNTRVFY uses this value as the OWNER for BIND PACKAGE and BIND PLAN statements. If this value is not specified, the primary authorization ID is used for BIND PACKAGE, BIND PLAN, and issued SQL statements.

PKOWNER(pkowner)

Specifies the default owner for binding the DSNTRVFY plan and package. If the INSTALL PKG OWNER field on panel DSNTPTPG is specified, the installation CLIST uses that value to set PKOWNER(pkowner) during the editing of DSNTIJRV. Otherwise, it uses the value specified in the ROUTINES CREATOR field on panel DSNTPTPG. If SYSINSTL is specified on panel DSNTPTPG, the installation CLIST adds a notice to the DSNTIJRV prolog to explain additional requirements.

If you use an authorization ID with the installation SYSOPR authority to execute job DSNTIJRV, the ID must be explicitly granted the MONITOR2
system privilege. If you set AUTHID to SYSINSTL for running the
SYSPROC.WLM_REFRESH stored procedure, you must also define SYSINSTL
in RACF.

Output

DSNTRVFY allocates the following required DD statements for output:

SYSPRINT
Contains messages generated during verification processing. Each processed
routine can generate one or more messages.

SYSTSPRT
DSNTRVFY writes bind output to SYSTSPRT if an error occurs during bind
processing.

The following JCL example is for validating a single routine:

```
//DSNTRVFY EXEC PGM=DSNTRVFY,
// PARM='DB2SSID(VA1A) ROUTINE(SYSPROC.DSNUTILS)
//DBRMLIB DD DSN=DB2A.SDSNDBRM,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(32000,(30,30)),DCB=(RECFM=VB,LRECL=133)
//SYSTSPRT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

SYSPROC.DSNWZP
SYSPROC.DSNACCOR
SYSPROC.DSNUTILS
SYSPROC.DSNUTILU
```

The following JCL example is for validating multiple routines:

```
//DSNTRVFY EXEC PGM=DSNTRVFY,
// PARM='DB2SSID(VA1A) ROUTINE(DD:SYSIN)' 
//DBRMLIB DD DSN=DB2A.SDSNDBRM,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(32000,(30,30)),DCB=(RECFM=VB,LRECL=133)
//SYSTSPRT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

SYSPROC.DSNWZP
SYSPROC.DSNACCOR
SYSPROC.DSNUTILS
SYSPROC.DSNUTILU
```

The following JCL example is for validating all routines:

```
//DSNTRVFY EXEC PGM=DSNTRVFY,COND=(4,LT),
// PARM=('DB2SSID(DSN) AUTHID(SYSINSTL) '
// 'PKOWNER(ADMFO01) ROUTINE(ALL)')

//DBRMLIB DD DSN=DB2A.SDSNDBRM,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(32000,(30,30)),DCB=(RECFM=VB,LRECL=133)
//SYSTSPRT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
```

Processing

DSNTRVFY completes the following validation:

- Verifies that the routine is defined to the SYSIBM.SYRSUTINES catalog.
- Verifies that the collection ID defined in the SYSIBM.SYRSUTINES catalog
  table matches the collection ID used to bind the package.
- Verifies that the routine is currently defined with the Db2-reserved WLM
  application environment name DSNWLM_NOT_ENABLED, indicating that it is
  not enabled to run on the current Db2 subsystem. If it is, then a warning
  message is issued and the validation is bypassed.
• Verifies if the routine can be executed on the current Db2 catalog mode.
• Verifies the APF authorization requirements.
• Verifies if the WLM application environment is defined and available for use.
• Verifies if the NUMTCB setting is valid or within the recommended range.
• Verifies if the required DD statements for some of the routines are defined.
• Verifies if the WLM environment assigned to Java routines is enabled for Java.
• Verifies security coexistence requirements (for example, some routines should run on only their own WLM application environment).
• Verifies that the selected routine can be called successfully. Certain routines such as DSNACICS, DSNAIMS, MQSeries functions, and most debugger routines are restricted to basic validation and are not called.

During the running of this job, the following message might be issued on the system console:
IEFC452I  DB2UDSMD - JOB NOT RUN - JCL ERROR

In most cases, this message is normal and can be disregarded. If you use IBM Optim Development Studio, see [Debugging stored procedures on Db2 for z/OS with Optim Development Studio, Part 2 (IBM developerWorks)] for information about DB2UDSMD.

If the validation is successful, the following message is issued:
DSNT027I DSNTRVFY VALIDATION FOR schema.routine-specific-name SUCCESSFUL

If the validation fails, messages are issued to explain the reasons for failure.
Examine these messages and make necessary adjustments before running the validation job again.

Input

DSNTRVFY requires the following DD statements for input:

DBRMLIB
Identifies the library where the DBRM for the DSNTRVFY program is located.
This is typically the prefix.SDSNDBRM target library.

SYSIN
Contains the list of Db2 routines. This is required when the validation type is for multiple routines (DD:SYSIN).

SYSUT1
Work data set required for bind processing.

Limitations

The following limitations apply to the DSNTRVFY program:
• The name of the routine in the form schema-name.routine-specific-name is limited to 72 characters, the maximum line length in JCL.
• Only Db2 12 Db2-supplied routines are supported for validation.
• Some routines, if enabled, receive basic validation to ensure that they are defined correctly and assigned to a valid WLM application environment. However, validation will not extend to calling these routines for one or more of the following reasons:
  – Optional system support such as CICS, IMS, or MQSeries is required.
– External setup on a remote server or workstation is required.
– A call could introduce or modify data unexpectedly in a production environment.

The Db2-supplied routines that are restricted to basic validation are:
– DB2DEBUG.CREATE_SESSION
– DB2DEBUG.DESTROY_SESSION
– DB2DEBUG.GET_REPORT
– DB2DEBUG.LIST_SESSION
– DB2DEBUG.PUT_COMMAND
– SYSFUN.SYSTS_ENCRYPT
– SYSFUN.XSLTRANSFORM
– SYSPROC.ADMIN_COMMAND_UNIX
– SYSPROC.ADMIN_JOB_CANCEL
– SYSPROC.ADMIN_JOB_FETCH
– SYSPROC.ADMIN_JOB_QUERY
– SYSPROC.ADMIN_JOB_SUBMIT
– SYSPROC.DBG_ENDSESSIONMANAGER
– SYSPROC.DBG_INITIALIZECLIENT
– SYSPROC.DBG_RECVCLIENTREQUESTS
– SYSPROC.DBG_SENDCLIENTREQUESTS
– SYSPROC.DBG_SENDCLIENTCOMMANDS
– SYSPROC.DBG_TERMINATECLIENT
– SYSPROC.DSNACICS
– SYSPROC.DSNAIMS
– SYSPROC.DSNAIMS2
– SYSPROC.DSNLEUSR
– SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY
– SYSPROC.SYSTS_ALTER
– SYSPROC.SYSTS_CREATE
– SYSPROC.SYSTS_DROP
– SYSPROC.SYSTS_RESTORE
– SYSPROC.SYSTS_START
– SYSPROC.SYSTS_STOP
– SYSPROC.SYSTS_TAKEOVER
– SYSPROC.SYSTS_UPDATE
– All MQSeries, web services, and SOAP functions (those with schema DB2MQ, DB2MQ1C, DB2MQ2C, or DB2XML)

Notes

Temporary stored procedures used by DSNTIJRV
DSNTRVFY creates, calls, and drops two temporary stored procedures. These temporary stored procedures are
SYSPROC.DSNWLM_INSTALL_INFOJ and
SYSPROC.DSNWLM_INSTALL_INFO. Both of these stored procedures exist only when DSNTRVFY is executing. DSNTRVFY calls
SYSPROC.DSNWLM_INSTALL_INFOJ to validate the WLM environment that is assigned to each Db2-supplied Java routine. DSNTRVFY calls
SYSPROC.DSNWLM_INSTALL_INFO to validate the WLM environment that is assigned to each Db2-supplied non-Java routine that runs in a WLM environment.

**Using DSNTIJRV to verify 64-bit Java routines**

Java on z/OS has two types of architectures: a 31-bit version and a 64-bit version. If your Java routines run in a 31-bit Java virtual machine (JVM), the startup procedure for the WLM environment in which the routines run must initialize a 31-bit environment. If you are running Java routines in a 64-bit JVM, the startup procedure for the WLM environment in which your Java routines run must initialize a 64-bit environment. The program that initializes a 31-bit WLM environment is DSNX9WLM. The program that initializes a 64-bit WLM environment is DSNX9WJM.

For example, you might use the Db2 core Java WLM environment, DSNWLM_JAVA, for your Java routines. By default, the DSNWLMJ startup procedure for DSNWLM_JAVA runs program DSNX9WLM. If your JAVA_HOME environment variable specifies the path to a 31-bit JVM, such as /usr/lpp/java/IBM/J8.0, you need to run program DSNX9WLM to initialize the WLM environment. However, if your JAVA_HOME variable specifies a path to a 64-bit JVM, such as /usr/lpp/java/8.0_64, you need to run program DSNX9WJM to initialize the WLM environment. If you switch to a 64-bit JVM for your Java stored procedures, but do not update the initialization program to DSNX9WJM, DSNTIJRV fails.

**Related concepts:**

- WLM address space startup procedure for Java routines (Db2 Application Programming for Java)

**Migration step 32: Verify successful migration to Db2 12**

You can run the sample applications to verify that your migration process was successful.

**Before you begin**

Before you can run the Db2 12 sample applications, you must activate function level 500 or higher, as described in Migration step 29: Activate function level 500.

Until you complete this task, continue to use sample applications such as DSNTIAUL, DSNTEP2, and DSNTEP4 from Db2 11.

**Procedure**

To verify successful migration to Db2 12:

- Tailor and run jobs for the sample applications, as described in Verifying successful Db2 12 installation or migration with sample applications. You can select the phases that you need to run based on the attachment facilities you installed, the languages you use, and whether the sample objects exist.

**Related concepts:**

- Dynamic SQL programs: DSNTIAAD, DSNTEP2, DSNTIAUL

**Related tasks:**

Migrating Db2 to Db2 12
Falling back to Db2 11

Falling back is the process of returning to Db2 11 after migrating your catalog and directory to Db2 12. You can fall back to Db2 11 only if Db2 12 is at function level 100.

Before you begin

Fallback to Db2 11 is not possible after the activation of function level 500 or higher in Db2 12.

Before you fall back to your previous Db2 version, consider the implications of doing so. For details, see “Implications of falling back” on page 559.

About this task

You can fall back to Db2 11 only after successfully migrating the catalog to Db2 12 by using job DSNTIJTC. However, you cannot fall back to Db2 11 after you activate function level 500 or higher.

Fall back if you have a severe error while operating Db2 12 and you want to return to operation on the previous version. After fallback, the catalog remains a Db2 12 catalog.

Remigrating is the process of returning to Db2 12 after falling back.

Procedure

To fall back to Db2 11, complete the following steps:
1. Stop Db2 12.
2. Run job DSNTIJFV to rename the cataloged procedures and reactivate Db2 11 code.
3. Reconnect TSO, IMS, and CICS to Db2 11.
4. Start Db2 11.
5. Run the Db2 11 installation verification jobs (DSNTEJxx).

What to do next

You can save your Db2 12 TSO logon procedures and JCL for remigration to Db2 12.

Related tasks:
- Installation step 16: Tailor the Db2 catalog: DSNTIJTC
- Migration step 29: Activate function level 500
- Remigrating

Fallback step 1: Stop Db2 12 activity

Before you can fall back, you must stop Db2 12 activity and ensure that no recovery is required on system databases.

Procedure

To stop Db2 12 work:
1. Issue the following command:
The QUIESCE keyword allows Db2 to complete processing of currently executing programs. This activity might require some processing time.

2. Issue the following command:

   -DSN1 STOP DB2 MODE(QUIESCE)

   -DSN1 START DB2 ACCESS(MAINT)

   This command allows only the installation-defined system administrators and system operators to access Db2.
   If Db2 does not start properly, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. Check to see that you started Db2 with the correct subsystem parameter load module.

3. Ensure that all work is complete.
   - Ensure that no units of recovery remain. Issue the following command:

     -DSN1 DISPLAY THREAD(*) TYPE(*)

   Then use RECOVER INDOUBT for any indoubt threads.
   - Ensure that no utility work remains. Issue the following command:

     -DSN1 DISPLAY UTILITY(*)

   Then, either allow utilities to complete before proceeding, or stop all utility processing with the following command:
   -DSN1 TERM UTILITY(*)

   - Ensure that no table spaces and index spaces in the Db2 directory (DSNDB01) or the Db2 catalog (DSNDB06) have write error ranges or deferred restart states. Issue the following command:

     -DSN1 DISPLAY DATABASE(DSNDB01) SPACENAM(*) RESTRICT
     -DSN1 DISPLAY DATABASE(DSNDB06) SPACENAM(*) RESTRICT
A user with installation-defined system administrator or system operator authority also must enter this command. Recover any table spaces and index spaces with write error range or deferred restart states.

4. To stop Db2, issue the following command:

   -DSN1 STOP DB2 MODE(QUIESCE)

A user with SYSADM or SYSOPR authority also must enter this command. If IRLM does not stop automatically when Db2 stops, stop IRLM manually. To stop IRLM, issue the following command, where irlmproc is the name you assigned to the IRLM startup procedure:

   STOP irlmproc

**Fallback step 2: Reactivate Db2 11 code: DSNTIJFV**

After you stop your Db2 12 activity, continue the fallback process by using job DSNTIJFV to reactivate the code of your previous Db2 version.

**About this task**

Job DSNTIJFV renames procedures to activate Db2 11 and deactivate Db2 12.

**Procedure**

To reactivate the code of your previous Db2 version:

1. (Optional) Tailor job DSNTIJFV.

   SYS1.PROCLIB is the default target for JCL procedures. Add statements to rename other procedures, such as your IMS, CICS, TSO logon procedures, and batch procedures.

   You might want two sets of procedures, such as DSN1xxxx and DSN2xxxx, at all times, with an alias for the current release level.

2. Run job DSNTIJFV.

3. Ensure that job DSNTIJFV completed successfully.

   - If DSNTIJFV runs successfully, it produces a return code of 0. Check that all renames executed successfully.
   - If DSNTIJFV fails or abends, rerun only the renames that failed. If some of the procedures already exist, check carefully to ensure that procedures for the two releases are not mixed.

**Fallback step 3: Reconnect TSO, IMS, and CICS to Db2 11**

After you reactivate your Db2 11 code during fallback, reconnect TSO, IMS, and CICS to Db2 11.
Procedure

To reconnect TSO, IMS, and CICS to Db2:

1. Re-establish your Db2 11 TSO logon procedures and JCL.
2. For CICS connections, complete one of the following actions:
   - If you overwrote the load module during migration to Db2 12, reassemble the resource control table (RCT) with the Db2 11 libraries.
   - If you did not overwrite the load module, change the STEPLIB statements in Db2 CICS jobs so that they refer to the Db2 11 libraries.
3. For IMS connections, change the STEPLIB statements in Db2 IMS jobs and address spaces so that they refer to the Db2 11 libraries.

Fallback step 4: Start Db2 11

During fallback, you can start Db2 11 after you re-establish your Db2 11 TSO logon procedures, CICS connections, and IMS connections.

Procedure

To start Db2 11:

1. Start the IRLM. If you have not requested that Db2 automatically start the IRLM, start it before you start Db2. Use the following command, where irlmproc is the name that you assigned to the IRLM startup procedure:
   
   START irlmproc
   
   This is the value that you specified for the PROC NAME option on installation panel DSNTIPI.
   
   If you specified YES for the AUTO START option on installation panel DSNTIPI, Db2 starts the IRLM automatically.

2. Start Db2 from the z/OS console by using the following command:

   -DSN1 START DB2,PARM(DSNZPxxx)

   In this command, -DSN1 is the subsystem command prefix that you defined for Db2, and DSNZPxxx is the name of the Db2 11 subsystem parameter module. If you used the default name, DSNZPROM, you can omit the PARM parameter.
   
   If Db2 starts successfully, two to five address spaces also start. These address spaces are ssnmMSTR and ssnmDBM1, and possibly ssnmDIST and irlmproc, where ssnm is the Db2 subsystem name and irlmproc is the IRLM procedure name.
   
   If Db2 starts successfully, the series of restart messages that you receive concludes with these two messages:

   DSNR002I  RESTART COMPLETED
   DSN9022I  DSNYASCP '-DSN1 START DB2' NORMAL COMPLETION

3. Complete one of the following actions:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you have done distributed processing with your Db2 12 subsystem</td>
<td>Check message DSNR005I for the number of indoubt threads after you start Db2. If you find no indoubt threads, continue falling back as if you had not done any distributed processing. If you find indoubt threads, issue the following command: DSN1 DISPLAY THREAD(*) TYPE(INDOUBT) If the number of indoubt threads that are reported in the DSNV408I messages is equal to the number of threads that are reported in the DSNR005I message, continue falling back as if you had not done any distributed processing. If fewer indoubt threads are reported by DSNV408I messages than in message DSNR005I, proceed as follows: 1. Stop Db2 11. 2. Determine which units of work are incomplete by scanning the Db2 recovery log with the Db2 12 DSN1LOGP utility. Use the SUMMARY option of this utility. 3. Examine the DSN1LOGP output to find all the DSN1162I messages that have a COORDINATOR name in a remote location. Each of these messages identify an indoubt DBAT. Record the LUWID that is displayed in each message. 4. Decide whether to commit or abort each indoubt DBAT. One way to do this is by contacting the COORDINATOR location. If it is another Db2 subsystem, use the DISPLAY THREAD command to help you decide. 5. If you have not already done so during migration, apply the fallback PTF that is supplied with Db2 12. 6. Start Db2 11 again. 7. Issue the RECOVER INDOUBT ACTION(correct decision) LUWID(luwid) command to resolve each indoubt DBAT.</td>
</tr>
<tr>
<td>If you have not done distributed processing with your Db2 12 subsystem</td>
<td>Check outstanding restrictions after you start Db2. Identify databases whose uses are restricted by issuing the following command: DSN1 DISPLAY DATABASE(<em>) SPACENAM(</em>) RESTRICT You can start some of these databases at this time.</td>
</tr>
</tbody>
</table>

4. If Db2 does not start properly, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. A common cause of startup failure is that the BSDS does not match the subsystem parameter values; ensure that the startup procedure is pointing to the correct BSDS and subsystem parameter. Also, check that the subsystem parameter member that you specified (or is used by default) when you started Db2 is the one that job DSNTIJUZ built. Check the JCL for the Db2 startup procedure.

5. Optional: Start TSO. If you want to use the TSO SUBMIT command to do housekeeping and fallback verification, you must start TSO (if it is not already started).

Related tasks:
Fallback step 5: Verify fallback

After you start Db2 11, you must complete some of your own testing to determine if the fallback was successful.

About this task

You cannot run the Db2 12 samples on Db2 11.

Procedure

To verify fallback:

1. Test the IVP phase 2 applications:
   a. DSNTEJ2A: Perform all except the first two steps of job DSNTEJ2A. Expect a return code of 4 because table spaces DSN8D11U.NEWDEPT and DSN8D11U.NEWPHONE are placed in COPY-pending states.
   b. DSNTEJ2C: Issue only the RUN PROGRAM(DSN8BC3) PLAN(DSN8BH11) statement in step PH02CS04.
   c. DSNTEJ2D: Issue only the RUN PROGRAM(DSN8BD3) PLAN(DSN8BD11) statement in step PH02DS03.
   d. DSNTEJ2E: Issue only the RUN PROGRAM(DSN8BE3) PLAN(DSN8BE11) statement in step PH02ES04.
   e. DSNTEJ2F: Issue only the RUN PROGRAM(DSN8BF3) PLAN(DSN8BF11) statement in step PH02FS03.
   f. DSNTEJ2P: Run only step PH02PS05.
2. Test the IVP phase 3 applications. Do not run job DSNTEJ3C or DSNTEJ3P.
3. Run the Db2 11 sample applications.
4. Test your own applications.
5. Retry the problem for which you decided to fall back.

Related tasks:

- Migration step 28: Verify migration of Db2 11 IVP objects to Db2 12 (optional)

Implications of falling back

Before you fall back to your previous Db2 version, consider the implications of doing so.

Data sharing

If any member of a data sharing group needs to fall back, consider the implications for data sharing. For more information about falling back in a data sharing environment, see Falling back in a data sharing environment.

Frozen objects

Falling back does not undo changes that the migration process made to the catalog. Db2 uses the migrated catalog after fallback. Some objects in this catalog that have been affected by Db2 12 function might become frozen objects after fallback. Frozen objects are unavailable, and they are marked with the release dependency marker Q. If an object is marked with a release dependency, it remains marked forever.
The release dependency marker is listed in the IBMREQD column of catalog tables. The following table lists the objects that are frozen when you fall back to a previous release.

**Table 48. Objects that are frozen after fallback**

<table>
<thead>
<tr>
<th>RELEASE DEPENDENT MARK = Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Plans, packages, or views that use any new syntax, objects, or bind options</td>
</tr>
<tr>
<td>- DBRMs that are produced by a precompile in Db2 12 with a value of V12 for the NEWFUN option and when any new Db2 12 functions are being used</td>
</tr>
</tbody>
</table>

Plans and packages become frozen objects when they use new SQL syntax, use new BIND options and attributes, or reference frozen objects. When plans and packages become frozen objects, the automatic rebind process is adversely affected.

**GUPI** After falling back, you can determine if any of your objects are frozen by issuing the following statement:

```sql
SELECT * FROM catalog_table
WHERE IBMREQD = 'Q'
```

**Automatic rebind**

After fallback, if you specified YES or COEXIST for the ABIND subsystem parameter, plans or packages that are bound in Db2 12 are automatically rebound on their first execution in Db2 11. If you specified NO for the ABIND subsystem parameter, this automatic rebind fails.

The automatic rebind also fails if you try to use plans or packages that are frozen. To make available the plans and packages that were not automatically rebound on Db2 11, change the SQL statements or remove the reference to a frozen object, precompile the application programs, and explicitly bind the plans and packages on Db2 11.

**Administrative task scheduler tasks**

After fallback, all tasks that are managed by the administrative task scheduler in Db2 12 are not manageable by the administrative task scheduler in Db2 11 because of the higher task version. Use ADMIN_TASK_ADD in Db2 11 to delete and reinsert all tasks. You can use the ADMIN_TASK_LIST table user-defined function to read and store the characteristics of the tasks in a user table before you fall back.

**Related concepts:**

- Falling back and remigrating in a data sharing environment
- Automatic rebinds (Db2 Application programming and SQL)

**Remigrating**

Remigrating (returning to Db2 12 after falling back to Db2 11) is simpler than the initial migration process.
Before you begin

Before you remigrate, refer to Incompatible changes in Db2 12 (Db2 for z/OS What's New?) because many of those considerations apply to remigrations, too. Whether each consideration applies to your subsystem depends on the type of activity that took place on your Db2 11 subsystem after falling back.

Regardless of the ABIND subsystem parameter value, plans and packages continue to run on Db2 12 as bound on Db2 11, until you explicitly rebind them on Db2 12. It is best to do so only after you are certain that Db2 12 is stable and release coexistence and fallback are no longer a concern.

**Recommendation:** Avoid automatic rebinds during migration and remigration. After your remigrated system is stable, you can rebind your packages with PLANMGMT(EXTENDED). If a new package does not perform adequately, you can rebind with SWITCH(PREVIOUS) to return to the old package.

**Procedure**

To remigrate your subsystem to Db2 12:

1. Check the integrity of your Db2 11 catalog and directory:
   - Run the DSN1COPY utility with the CHECK option on your catalog table spaces.
   - Run the CHECK DATA utility.
   - Run the queries in member DSNTESQ of prefix.SDSNSAMP.
2. Optional: Ensure that your Db2 11 DSNTIJIC job has been modified to copy new Db2 12 catalog table spaces, and then use DSNTIJIC to take an image copy of your Db2 11 catalog. This step is not required, but it is recommended.
3. Stop Db2 11.
4. Reconnect TSO, IMS, and CICS to Db2. Re-establish your Db2 12 logon procedures and JCL, as well as your Db2 12 CICS and IMS connections.
5. Rebuild Db2 12 cataloged procedures. Rename the Db2 12 procedures that were renamed by job DSNTIJFV during fallback. If job DSNTIJFV was not run, you need to rerun job DSNTIJMV. Comment out step 1 (DSNTIMP), which defines Db2 12 to z/OS, and run the job. (You do not need to define Db2 12 to z/OS a second time.)
6. Start Db2 12. Ensure that you are using your Db2 12 subsystem parameter load module.
7. Take an image copy of the Db2 12 catalog by using the DSNTIJIC job.
8. Verify your Db2 12 system.

**Related tasks:**

- Migration step 2: Verify the integrity of Db2 table spaces (optional)
- Migration step 5: Take image copies of the directory and catalog: DSNTIJIC
- Migration step 28: Verify migration of Db2 11 IVP objects to Db2 12 (optional)
Chapter 7. Automating Db2 installation or migration with z/OS Management Facility

You can use z/OSMF to automate the process of installing or migrating to Db2 12 for z/OS.

Before you begin

You must use SMP/E to load the required Db2 libraries, as described in Loading the Db2 libraries. Running the installation CLIST requires the following Db2 load libraries:

- `prefix.SDSNCLST`
- `prefix.SDSNSPFM`
- `prefix.SDSNSPFP`
- `prefix.SDSNSAMP`
- `prefix.SDSNIVPD` (only if generating z/OSMF artifacts)

For migrations to Db2 12, this process uses new and modified panels for the installation CLIST. The panels reside in the Db2 12 SDSNSPFP library. You must make the panels available to TSO, ISPF, and the Db2 12 installation CLIST. For more information, see Tailoring Db2 12 installation and migration jobs with the CLIST.

About this task

By using z/OSMF, you can automate the process of migrating or installing Db2 by creating workflow definition and variable files that you can define one time for reuse across many different Db2 subsystems.

Related concepts:
- About workflows (z/OS Management Facility)

Related tasks:
- Activating Db2 function levels by using z/OSMF (Db2 for z/OS What's New?)

Installing and migrating Db2

Related reference:
- DSNTIPA1 or DSNTILA1: Main panel
- IBM z/OS Management Facility Online Help

Related information:
- Automating Db2 for z/OS Migration With z/OS Management Facility (white paper)

Automating Db2 installation by using z/OS Management Facility

You can use z/OSMF to automate the process of installing Db2 12 for z/OS subsystems.
Before you begin

You must have appropriate authority to complete this task. For details, see Required authorization for installation or migration.

You must use SMP/E to load the required Db2 libraries, as described in Loading the Db2 libraries. Running the installation CLIST requires the following Db2 load libraries:

- `prefix.SDSNCLST`
- `prefix.SDSNSPFM`
- `prefix.SDSNSPFP`
- `prefix.SDSNSAMP`
- `prefix.SDSN1VPD` (only if generating z/OSMF artifacts)

About this task

By using z/OSMF, you can automate the process of installing Db2 by creating workflow definition and variable files that you can define one time for reuse across many different Db2 subsystems.

Procedure

To install Db2 subsystems by using z/OSMF, use the following general approach:

1. Identify an existing Db2 subsystem that is the model for your new installations, or install one as described in Chapter 5, “Installing Db2,” on page 413.

2. Create the workflow definition files:

   a. On the existing Db2 subsystem that is the model for the new installations, run the DSNTXAZP tool to update the CLIST defaults input member, as described in Updating the CLIST defaults input member: job DSNTIJXZ.

      **Important**: The DSNTXAZP tool does not update the DSNTIDxx CLIST input member with current settings for the following parameters or panel fields. Update the values as you complete the CLIST, or modify the values directly in the DSNTIDxx member:

      Table 49. Settings not updated by the DSNTXAZP tool

      | CLIST Panel | Field Name (DSNTIDxx entry) |
      |-------------|-----------------------------|
      | DSNTIPA2    | The following fields:       |
      |             | CATALOG ALIAS (VCATALOG)    |
      |             | DEFINE CATALOG (VCATSTAT)   |
Table 49. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIPA3</td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>PERMANENT UNIT NAME</td>
</tr>
<tr>
<td></td>
<td>(VOLSDEVT)</td>
</tr>
<tr>
<td></td>
<td>TEMPORARY UNIT NAME</td>
</tr>
<tr>
<td></td>
<td>(TVOLSDEVT)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS1)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS2)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS3)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: DATACLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: MGMTCLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS7)</td>
</tr>
<tr>
<td>DSNTIPK</td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>WORK FILE DB (WKDBNAME)</td>
</tr>
<tr>
<td></td>
<td>SUBGRP ATTACH (SGRPATNM)</td>
</tr>
</tbody>
</table>
Table 49. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPG</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>ROUTINES CREATOR (AUTHID)</td>
</tr>
<tr>
<td></td>
<td>SEC DEF CREATOR (SECDEFID)</td>
</tr>
<tr>
<td></td>
<td>INSTALL SQL ID (INSQLID)</td>
</tr>
<tr>
<td></td>
<td>INSTALL PKG OWNER (INSPKOWN)</td>
</tr>
<tr>
<td></td>
<td>INSTALL GRANTEE(S) (INSGRLST)</td>
</tr>
<tr>
<td></td>
<td>IVP SQL ID (IVPSQLID)</td>
</tr>
<tr>
<td></td>
<td>IVP PACKAGE OWNER (IVPPKOWN)</td>
</tr>
<tr>
<td></td>
<td>IVP GRANTEE(S) (IVPGRLST)</td>
</tr>
<tr>
<td><strong>DSNTIPG1</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>INSTALL IC PREFIX (INSICPRE)</td>
</tr>
<tr>
<td></td>
<td>INSTALL DS PREFIX (INSDSPRE)</td>
</tr>
<tr>
<td></td>
<td>IC DATA SET UNIT NAME (ICSDSDEV)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td>(ICDSTPLT)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE DATA CLASS (ICDSDC)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE MGMT CLASS (ICDSMC)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE STOR CLASS (ICDSSC)</td>
</tr>
<tr>
<td><strong>DSNTIPH</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>COPY 1 NAME (BSDSNAM1)</td>
</tr>
<tr>
<td></td>
<td>COPY 2 NAME (BSDSNAM2)</td>
</tr>
<tr>
<td></td>
<td>Active logs: COPY 1 PREFIX (LOGSPRE1)</td>
</tr>
<tr>
<td></td>
<td>Active logs: COPY 2 PREFIX (LOGSPRE2)</td>
</tr>
<tr>
<td></td>
<td>Archive logs: COPY 1 PREFIX (ARCHPRE1)</td>
</tr>
<tr>
<td></td>
<td>Archive logs: COPY 2 PREFIX (ARCHPRE2)</td>
</tr>
<tr>
<td><strong>DSNTIPT</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>TEMP CLIST LIBRARY (NEWCL)</td>
</tr>
<tr>
<td></td>
<td>SAMPLE LIBRARY (NEWSAMP)</td>
</tr>
<tr>
<td></td>
<td>CLIST LIBRARY (OUTCL)</td>
</tr>
<tr>
<td></td>
<td>APPLICATION DBRM (DBRMLIB)</td>
</tr>
<tr>
<td></td>
<td>APPLICATION LOAD (RUNLIB)</td>
</tr>
<tr>
<td></td>
<td>DECLARATION LIBRARY (SRCLIB)</td>
</tr>
<tr>
<td></td>
<td>LINK LIST LIBRARY (DSNLINK)</td>
</tr>
<tr>
<td></td>
<td>LOAD LIBRARY (DSNLOAD)</td>
</tr>
<tr>
<td></td>
<td>JDBC AND SQLJ DLLS (DSNLOAD2)</td>
</tr>
<tr>
<td></td>
<td>MACRO LIBRARY (DSNMACS)</td>
</tr>
<tr>
<td></td>
<td>LOAD DISTRIBUTION (DSNALOAD)</td>
</tr>
<tr>
<td></td>
<td>EXIT LIBRARY (DSNEXIT)</td>
</tr>
<tr>
<td></td>
<td>DBRM LIBRARY (DSNDBRM)</td>
</tr>
<tr>
<td></td>
<td>IRLM LOAD LIBRARY (DSNIRLM)</td>
</tr>
<tr>
<td></td>
<td>IVP DATA LIBRARY (DSNIVPD)</td>
</tr>
<tr>
<td></td>
<td>INCLUDE LIBRARY (DSNCHDRS)</td>
</tr>
</tbody>
</table>
Table 49. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPU</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY (LERUN)</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY 2 (LERUN2)</td>
</tr>
<tr>
<td></td>
<td>IBM LE LINK EDIT LIB (LELKED)</td>
</tr>
<tr>
<td></td>
<td>IBM LE PRELINK MSG LIB (LEPLMSGL)</td>
</tr>
<tr>
<td></td>
<td>HIGH LEVEL ASSEMBLER LIB (IASMLIB)</td>
</tr>
<tr>
<td></td>
<td>C/CPP COMPILER MODULE (CPROG)</td>
</tr>
<tr>
<td></td>
<td>C/CPP COMPILER LIBRARY (CCOMP)</td>
</tr>
<tr>
<td></td>
<td>C/CPP HEADER LIBRARY (CHDERS)</td>
</tr>
<tr>
<td></td>
<td>C/370 COMPILER MESSAGES (CMGS)</td>
</tr>
<tr>
<td></td>
<td>C LE PRELINKER PHASE (CPRlk)</td>
</tr>
<tr>
<td></td>
<td>CPP LE PRELINKER PHASE (CPPPRLK)</td>
</tr>
<tr>
<td></td>
<td>CPP CLASS LIB HEADERS (CPHDRS)</td>
</tr>
<tr>
<td></td>
<td>CPP AUTO CALL LIBRARY (CPPAUTCL)</td>
</tr>
<tr>
<td></td>
<td>CPP CLASS LIBRARY (CPPCLASS)</td>
</tr>
<tr>
<td></td>
<td>CPP PROCEDURE LIBRARY (CPPPROC)</td>
</tr>
<tr>
<td><strong>DSNTIPU1</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>COBOL COMPILER LIBRARY (ICOBCOMP)</td>
</tr>
<tr>
<td></td>
<td>COBOL LE PRELINKER PHASE (ICOBPRlk)</td>
</tr>
<tr>
<td></td>
<td>FORTRAN COMPILER LIBRARY (FORTCOMP)</td>
</tr>
<tr>
<td></td>
<td>FORTRAN LINK EDIT LIB (FORTLINK)</td>
</tr>
<tr>
<td></td>
<td>PL/I COMPILER LIBRARY (PLICOMP)</td>
</tr>
<tr>
<td></td>
<td>PL/I LE PRELINKER PHASE (PLIPRLK)</td>
</tr>
<tr>
<td><strong>DSNTIPW</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>SYSTEM MACLIB (MACLIB)</td>
</tr>
<tr>
<td></td>
<td>SYSTEM PROCEDURES (PROCCLIB)</td>
</tr>
<tr>
<td></td>
<td>SORT LIBRARY (SORTLIB)</td>
</tr>
<tr>
<td></td>
<td>IMS RESLIB (RESLIB)</td>
</tr>
<tr>
<td></td>
<td>ISPF ISPLINK MODULE (ISPLINK)</td>
</tr>
<tr>
<td></td>
<td>GDDM MACLIB (GDDMMAACS)</td>
</tr>
<tr>
<td></td>
<td>GDDM LOAD MODULES (GDDMLOAD)</td>
</tr>
<tr>
<td></td>
<td>CICS LOAD LIBRARY (CICSLOAD)</td>
</tr>
<tr>
<td></td>
<td>CICS MACRO LIBRARY (CICSMAC)</td>
</tr>
<tr>
<td></td>
<td>CICS COBOL LIBRARY (CICSCOB)</td>
</tr>
<tr>
<td></td>
<td>CICS PL/I LIBRARY (CICSPLI)</td>
</tr>
<tr>
<td></td>
<td>CICS EXCI LIBRARY (CICSEXCI)</td>
</tr>
</tbody>
</table>
### Table 49. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPD</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>DATABASES (NUMDATAB)</td>
</tr>
<tr>
<td></td>
<td>TABLES (NUMTABLE)</td>
</tr>
<tr>
<td></td>
<td>COLUMNS (UMCOLUM)</td>
</tr>
<tr>
<td></td>
<td>VIEWS (NUMVIEWS)</td>
</tr>
<tr>
<td></td>
<td>TABLE SPACES (NUMTABSP)</td>
</tr>
<tr>
<td></td>
<td>PLANS (NUMPLANS)</td>
</tr>
<tr>
<td></td>
<td>PLAN STATEMENTS (NUMSTMTS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGES (NUMPKGS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGE STATEMENTS (NUMPKSTS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGE Lists (NUMPLPL)</td>
</tr>
<tr>
<td></td>
<td>EXECUTED STMTS (NUMSTMTE)</td>
</tr>
<tr>
<td><strong>DSNTIPS</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>4K SORT PRIMARY SPACE (S4KSPAC)</td>
</tr>
<tr>
<td></td>
<td>4K SORT WORK TS’S (S4KNOTS)</td>
</tr>
<tr>
<td></td>
<td>4K SORT WORK SEG SIZE (S4KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>32K SORT PRIMARY SPACE (S32KSPAC)</td>
</tr>
<tr>
<td></td>
<td>32K SORT WORK TS’S (S32KNOTS)</td>
</tr>
<tr>
<td></td>
<td>32K SORT WORK SEG SIZE (S32KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP PRIMARY SPACE (T4KSPAC)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP WORK TS’S (T4KNOTS)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP WORK SEG SIZE (T4KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>32K TEMP PRIMARY SPACE (T32KSPAC)</td>
</tr>
<tr>
<td></td>
<td>32K TEMP WORK TS’S (T32KNOTS)</td>
</tr>
<tr>
<td><strong>DSNTIPM</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>SUBSYSTEM NAME (MVSNMAME)</td>
</tr>
<tr>
<td></td>
<td>COMMAND PREFIX (MVSPREF)</td>
</tr>
<tr>
<td><strong>DSNTIPR</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>DB2 LOCATION NAME (DDLOC)</td>
</tr>
<tr>
<td></td>
<td>DB2 NETWORK LUNAME (DDLUNAM)</td>
</tr>
<tr>
<td><strong>DSNTIPS</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>DRDA PORT (DRDAAPRT)</td>
</tr>
<tr>
<td></td>
<td>RESYNC PORT (REYPRT)</td>
</tr>
<tr>
<td><strong>DSNTIPX</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>WLMENV (WLMENV)</td>
</tr>
<tr>
<td><strong>DSNTIPR1</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>All fields (WLMENVx).</td>
</tr>
<tr>
<td><strong>DSNTIPRA</strong></td>
<td>On panels DSNTIPRB - DSNTIPRQ:</td>
</tr>
<tr>
<td></td>
<td>All WLM ENV NAME fields</td>
</tr>
<tr>
<td></td>
<td>(RT_xxxxWLM)</td>
</tr>
<tr>
<td><strong>DSNTIPY</strong></td>
<td>All numbered job card information fields.</td>
</tr>
</tbody>
</table>
b. Run the installation CLIST, and specify the following field values of panel DSNTIPA1: USE Z/OSMF WORKFLOW ===> YES and INSTALL TYPE ===> INSTALL. The CLIST runs with modified panels and extra panels to support generation of a z/OSMF workflow.

**Tip:** You can specify an HFS or zFS directory for storing the z/OSMF artifacts. Specify the directory in the PATH FOR Z/OSMF ARTIFACTS field on panel DSNTIPM1. For more information about the specific panels, see Installation panels for installing or migrating with z/OSMF.

c. Complete the CLIST as described in Tailoring Db2 12 installation and migration jobs with the CLIST. As you complete the CLIST, select each installation parameter that you want to include in the input variable file for the z/OSMF workflow.

You can more easily change values in the input variable file when you reuse the workflow for different subsystems. For each field that you do not select, the value that you specify in the panel is hardcoded into the workflow definition file.

**Tip:** To reduce the size and difficulty of managing your input variable files, choose only input fields that are likely to be different across different Db2 subsystems in your environment.

d. If you are installing multiple Db2 subsystems, run the CLIST again and specify the following field values on panel DSNTILA1: USE Z/OSMF WORKFLOW ===> YES and INSTALL TYPE ===> UPDATE. The CLIST runs with an abbreviated set of panels to create input variable files for subsystems after the first subsystem.

Repeat this step for each subsystem after the first.

3. Create the workflow instances in z/OSMF.

   a. Specify the path of the workflow definition file and the input variable file. The path of the input variable file is optional, but it is best to always specify it. Otherwise, z/OSMF prompts you to provide the variable values during execution, preventing full automation of the process.

   b. Specify a name for the workflow instance.

   **Tip:** Specify a name that describes the purpose of the workflow and includes the name of the target system. For example, you might use the following name: V10 to V11 CM non-data-sharing - DB2A, where “DB2A” is the name of the target subsystem.

   c. Assign all workflow steps to the same person to fully automate the migration process.

   d. Validate the JCL job and the substituted input variables. You can find the input variable files and the JCL jobs with the substituted input variables on the Perform tab for the workflow in z/OSMF.

      1) Review the input variables. You can edit the values of input variables during your review, but any changes that you make apply only to this workload instance only.

      2) In the Create JOB statement step, z/OSMF prompts you to review and update the JOB statement JCL. You can specify appropriate CLASS or MSGCLASS values.
If installation or migration step requires authorization, you can also specify the USERNAME and PASSWORD. For detailed information about required authorizations see Required authorization for installation or migration.

3) Review the JCL including the JOB statement. You can click Edit JCL to make changes. However, any changes that you make apply only to this workflow instance. It is best to make any changes that apply to all instances in the JCL template before you create the instances.

4. Start running the workflow steps. You can run the workflow steps in z/OSMF, or by embedding z/OSMF REST services in your own automation programs.

a. After any prerequisite manual steps are complete, select Automatically perform the selected step, and all subsequent steps... on the Perform Automated Step page. z/OSMF runs all subsequent steps in order until all steps complete successfully, or until an error is encountered.

b. If errors are encountered, check the job output on the z/OSMF Status tab. The maximum return code that a step tolerates is defined the workload definition file, and it might be a return code other than 0. The maximum return code for most steps is 0 or 4.

Related concepts:

- About workflows (z/OS Management Facility)

Related tasks:

- Installing Db2
- Migrating Db2 to Db2 12

Related reference:

- DSNTIPA1 or DSNITLA1: Main panel
- IBM z/OS Management Facility Online Help

Automating Db2 migration by using z/OS Management Facility

You can use z/OSMF to automate the process of migrating Db2 subsystems to Db2 12 for z/OS.

Before you begin

For migrations to Db2 12, this process uses new and modified panels for the installation CLIST. The panels reside in the Db2 12 SDSNSPFPP library. You must make the panels available to TSO, ISPF, and the Db2 12 installation CLIST. For more information, see Tailoring Db2 12 installation and migration jobs with the CLIST.

You must have appropriate authority to complete this task. For details, see Required authorization for installation or migration.

You must use SMP/E to load the required Db2 libraries, as described in Loading the Db2 libraries. Running the installation CLIST requires the following Db2 load libraries:

- `prefix.SDSNCLST`
- `prefix.SDSNSPFM`
- `prefix.SDSNSPFPP`
- `prefix.SDSNSAMP`
- `prefix.SDSNIVPD` (only if generating z/OSMF artifacts)
Read the white paper about this process Automating Db2 for z/OS Migration With z/OS Management Facility (white paper).

**About this task**

By using z/OSMF, you can automate the process of migrating Db2 by creating workflow definition and variable files that you can define one time for reuse across many different Db2 subsystems.

In summary, you run the installation CLIST with certain options to generate the artifacts that define the z/OSMF workflow, including workflow definition files and input variable files. For more information about specific sample workflow artifacts, see Sample workflow artifacts for installation and migration with z/OS Management Facility.

The number of workflow definition and input variable files that you create depends on the environment that you are migrating. You run the CLIST again to generate each workflow definition file. For example, to migrate a single Db2 subsystem, only DSNTIWM is required. For data sharing groups, you must also create the DSNTIWMD workflow definition file to migrate members other than the first member to Db2 12.

Each workflow can use one or more input variable files. Each time that you run the CLIST to generate the workflow artifacts, subsystem parameters that you select are included in the generated input variable file. You can more easily change values in the input variable file when you reuse the workflow for different subsystems. Values for fields that you do not select are hardcoded into the workflow definition file. You might also create extra input variable files to support the efficient migration of several subsystems in your environment.

**Procedure**

To migrate to Db2 12 by using z/OSMF, use the following general approach:

1. Prepare for creating workflow definition files by running the DSNTXAZP tool to update the CLIST defaults input member, as described in Updating the CLIST defaults input member: job DSNTIJXZ.

   **Important:** The DSNTXAZP tool does not update the DSNTIDxx CLIST input member with current settings for the following parameters or panel fields. Update the values as you complete the CLIST, or modify the values directly in the DSNTIDxx member:

   **Table 50. Settings not updated by the DSNTXAZP tool**

```
<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIIPA2</td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>CATALOG ALIAS (VCATALOG)</td>
</tr>
<tr>
<td></td>
<td>DEFINE CATALOG (VCATSTAT)</td>
</tr>
</tbody>
</table>
```

Chapter 7. Automating Db2 installation or migration with z/OS Management Facility  571
<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPA3</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>PERMANENT UNIT NAME</td>
</tr>
<tr>
<td></td>
<td>(VOLSDEVVT)</td>
</tr>
<tr>
<td></td>
<td>TEMPORARY UNIT NAME</td>
</tr>
<tr>
<td></td>
<td>(TVOLSDEVVT)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS1)</td>
</tr>
<tr>
<td></td>
<td>CLIST ALLOCATION: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS1)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS2)</td>
</tr>
<tr>
<td></td>
<td>NON-VSAM DATA: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS2)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS3)</td>
</tr>
<tr>
<td></td>
<td>VSAM CATALOG, ...: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS3)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: DATACLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: MGMTCLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 1, BSDS 2: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS6)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: VOL/SER</td>
</tr>
<tr>
<td></td>
<td>(VOLSDAT7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: DATA CLASS</td>
</tr>
<tr>
<td></td>
<td>(DATACLS7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: MGMT CLASS</td>
</tr>
<tr>
<td></td>
<td>(MGMTCLS7)</td>
</tr>
<tr>
<td></td>
<td>LOG COPY 2, BSDS 1: STOR CLASS</td>
</tr>
<tr>
<td></td>
<td>(STORCLS7)</td>
</tr>
<tr>
<td><strong>DSNTIPK</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>WORK FILE DB (WKDBNAME)</td>
</tr>
<tr>
<td></td>
<td>SUBGRP ATTACH (SGRPATNM)</td>
</tr>
</tbody>
</table>
Table 50. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIPG</td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>ROUTINES CREATOR (AUTHID)</td>
</tr>
<tr>
<td></td>
<td>SEC DEF CREATOR (SECDEFID)</td>
</tr>
<tr>
<td></td>
<td>INSTALL SQL ID (INSQLID)</td>
</tr>
<tr>
<td></td>
<td>INSTALL PKG OWNER (INSPKOWN)</td>
</tr>
<tr>
<td></td>
<td>INSTALL GRANTEE(S) (INSGRLST)</td>
</tr>
<tr>
<td></td>
<td>IVP SQL ID (IVPSQLID)</td>
</tr>
<tr>
<td></td>
<td>IVP PACKAGE OWNER (IVPPKOWN)</td>
</tr>
<tr>
<td></td>
<td>IVP GRANTEE(S) (IVPGRLST)</td>
</tr>
<tr>
<td>DSNTIPG1</td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>INSTALL IC PREFIX (INSCIPE)</td>
</tr>
<tr>
<td></td>
<td>INSTALL DS PREFIX (INSDSPE)</td>
</tr>
<tr>
<td></td>
<td>IC DATA SET UNIT NAME (ICDSDS)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td>(ICDSTPLT)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE DATA CLASS (ICSDC)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE MGMT CLASS (ICSDMC)</td>
</tr>
<tr>
<td></td>
<td>IC TEMPLATE STOR CLASS (ICDSSC)</td>
</tr>
<tr>
<td>DSNTIPH</td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>COPY 1 NAME (BSDSNAM1)</td>
</tr>
<tr>
<td></td>
<td>COPY 2 NAME (BSDSNAM2)</td>
</tr>
<tr>
<td></td>
<td>Active logs: COPY 1 PREFIX (LOGSPRE1)</td>
</tr>
<tr>
<td></td>
<td>Active logs: COPY 2 PREFIX (LOGSPRE2)</td>
</tr>
<tr>
<td></td>
<td>Archive logs: COPY 1 PREFIX</td>
</tr>
<tr>
<td></td>
<td>(ARCHPRE1)</td>
</tr>
<tr>
<td></td>
<td>Archive logs: COPY 2 PREFIX</td>
</tr>
<tr>
<td></td>
<td>(ARCHPRE2)</td>
</tr>
<tr>
<td>DSNTIPT</td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>TEMP CLIST LIBRARY (NEWCL)</td>
</tr>
<tr>
<td></td>
<td>SAMPLE LIBRARY (NEWSAMP)</td>
</tr>
<tr>
<td></td>
<td>CLIST LIBRARY (OUTCL)</td>
</tr>
<tr>
<td></td>
<td>APPLICATION DBRM (DBRMLIB)</td>
</tr>
<tr>
<td></td>
<td>APPLICATION LOAD (RUNLIB)</td>
</tr>
<tr>
<td></td>
<td>DECLARATION LIBRARY (SRCLIB)</td>
</tr>
<tr>
<td></td>
<td>LINK LIST LIBRARY (DSNLINK)</td>
</tr>
<tr>
<td></td>
<td>LOAD LIBRARY (DSNLOAD)</td>
</tr>
<tr>
<td></td>
<td>JDBC AND SQLJ DLLS (DSNLOD2)</td>
</tr>
<tr>
<td></td>
<td>MACRO LIBRARY (DSNMACS)</td>
</tr>
<tr>
<td></td>
<td>LOAD DISTRIBUTION (DSNALOAD)</td>
</tr>
<tr>
<td></td>
<td>EXIT LIBRARY (DSNEXT)</td>
</tr>
<tr>
<td></td>
<td>DBRM LIBRARY (DSNDBRM)</td>
</tr>
<tr>
<td></td>
<td>IRLM LOAD LIBRARY (DSNRILM)</td>
</tr>
<tr>
<td></td>
<td>IVP DATA LIBRARY (DSNIVPD)</td>
</tr>
<tr>
<td></td>
<td>INCLUDE LIBRARY (DNSCHDRS)</td>
</tr>
<tr>
<td>CLIST Panel</td>
<td>Field Name (DSNTIDxx entry)</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td><strong>DSNTIPU</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY (LERUN)</td>
</tr>
<tr>
<td></td>
<td>IBM LE RUNTIME LIBRARY 2 (LERUN2)</td>
</tr>
<tr>
<td></td>
<td>IBM LE LINK EDIT LIB (LELKED)</td>
</tr>
<tr>
<td></td>
<td>IBM LE PRELINK MSG LIB (LEPLMSGL)</td>
</tr>
<tr>
<td></td>
<td>HIGH LEVEL ASSEMBLER LIB (IASMLIB)</td>
</tr>
<tr>
<td></td>
<td>C/CPP COMPILER MODULE (CPROG)</td>
</tr>
<tr>
<td></td>
<td>C/CPP COMPILER LIBRARY (CCOMP)</td>
</tr>
<tr>
<td></td>
<td>C/CPP HEADER LIBRARY (CHDRS)</td>
</tr>
<tr>
<td></td>
<td>C/370 COMPILER MESSAGES (CMGS)</td>
</tr>
<tr>
<td></td>
<td>C LE PRELINKER PHASE (CPRLK)</td>
</tr>
<tr>
<td></td>
<td>CPP LE PRELINKER PHASE (CPPPRLK)</td>
</tr>
<tr>
<td></td>
<td>CPP CLASS LIB HEADERS (CPPHDRS)</td>
</tr>
<tr>
<td></td>
<td>CPP AUTO CALL LIBRARY (CPPAUTCL)</td>
</tr>
<tr>
<td></td>
<td>CPP CLASS LIBRARY (CPPCLASS)</td>
</tr>
<tr>
<td></td>
<td>CPP PROCEDURE LIBRARY (CPPPROC)</td>
</tr>
<tr>
<td><strong>DSNTIPU1</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>COBOL COMPILER LIBRARY (ICOBCOMP)</td>
</tr>
<tr>
<td></td>
<td>COBOL LE PRELINKER PHASE (ICOBPRLK)</td>
</tr>
<tr>
<td></td>
<td>FORTRAN COMPILER LIBRARY (FORTCOMP)</td>
</tr>
<tr>
<td></td>
<td>FORTRAN LINK EDIT LIB (FORTLINK)</td>
</tr>
<tr>
<td></td>
<td>PL/1 COMPILER LIBRARY (PLICOMP)</td>
</tr>
<tr>
<td></td>
<td>PL/1 LE PRELINKER PHASE (PLIPRLK)</td>
</tr>
<tr>
<td><strong>DSNTIPW</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>SYSTEM MACLIB (MACLIB)</td>
</tr>
<tr>
<td></td>
<td>SYSTEM PROCEDURES (PROCLIB)</td>
</tr>
<tr>
<td></td>
<td>SORT LIBRARY (SORTLIB)</td>
</tr>
<tr>
<td></td>
<td>IMS RESLIB (RESLIB)</td>
</tr>
<tr>
<td></td>
<td>ISPF ISPLINK MODULE (ISPOLOAD)</td>
</tr>
<tr>
<td></td>
<td>GDDM MACLIB (GDDMMACBS)</td>
</tr>
<tr>
<td></td>
<td>GDDM LOAD MODULES (GDDMLOAD)</td>
</tr>
<tr>
<td></td>
<td>CICS LOAD LIBRARY (CICSMOD)</td>
</tr>
<tr>
<td></td>
<td>CICS MACRO LIBRARY (CICSMAC)</td>
</tr>
<tr>
<td></td>
<td>CICS COBOL LIBRARY (CICSCOB)</td>
</tr>
<tr>
<td></td>
<td>CICS PL/1 LIBRARY (CICSPLI)</td>
</tr>
<tr>
<td></td>
<td>CICS EXCI LIBRARY (CICSEXCI)</td>
</tr>
</tbody>
</table>
Table 50. Settings not updated by the DSNTXAZP tool (continued)

<table>
<thead>
<tr>
<th>CLIST Panel</th>
<th>Field Name (DSNTIDxx entry)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNTIPD</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>DATABASES (NUMDATAB)</td>
</tr>
<tr>
<td></td>
<td>TABLES (NUMTABLE)</td>
</tr>
<tr>
<td></td>
<td>COLUMNS (NUMCOLUM)</td>
</tr>
<tr>
<td></td>
<td>VIEWS (NUMVIEWS)</td>
</tr>
<tr>
<td></td>
<td>TABLE SPACES (NUMTABSP)</td>
</tr>
<tr>
<td></td>
<td>PLANS (NUMPLANS)</td>
</tr>
<tr>
<td></td>
<td>PLAN STATEMENTS (NUMSTMTS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGES (NUMPKGS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGE STATEMENTS (NUMPKSTS)</td>
</tr>
<tr>
<td></td>
<td>PACKAGE LISTS (NUMPLPL)</td>
</tr>
<tr>
<td></td>
<td>EXECUTED STMTS (NUMSTMTE)</td>
</tr>
<tr>
<td><strong>DSNTIP9</strong></td>
<td>All fields:</td>
</tr>
<tr>
<td></td>
<td>4K SORT PRIMARY SPACE (S4KSPAC)</td>
</tr>
<tr>
<td></td>
<td>4K SORT WORK TS’S (S4KNOTS)</td>
</tr>
<tr>
<td></td>
<td>4K SORT WORK SEG SIZE (S4KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>32K SORT PRIMARY SPACE (S32KSPAC)</td>
</tr>
<tr>
<td></td>
<td>32K SORT WORK TS’S (S32KNOTS)</td>
</tr>
<tr>
<td></td>
<td>32K SORT WORK SEG SIZE (S32KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP PRIMARY SPACE (T4KSPAC)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP WORK TS’S (T4KNOTS)</td>
</tr>
<tr>
<td></td>
<td>4K TEMP WORK SEG SIZE (T4KSGSZ)</td>
</tr>
<tr>
<td></td>
<td>32K TEMP PRIMARY SPACE (T32KSPAC)</td>
</tr>
<tr>
<td></td>
<td>32K TEMP WORK TS’S (T32KNOTS)</td>
</tr>
<tr>
<td><strong>DSNTIPM</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>SUBSYSTEM NAME (MVSNAME)</td>
</tr>
<tr>
<td></td>
<td>COMMAND PREFIX (MVSPREF)</td>
</tr>
<tr>
<td><strong>DSNTIPR</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>DB2 LOCATION NAME (DDLLOC)</td>
</tr>
<tr>
<td></td>
<td>DB2 NETWORK LUNAME (DDLUNAM)</td>
</tr>
<tr>
<td><strong>DSNTIP5</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>DRDA PORT (DRDAPRT)</td>
</tr>
<tr>
<td></td>
<td>RESYNC PORT (RESYNPRT)</td>
</tr>
<tr>
<td><strong>DSNTIPX</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>WLMENV (WLMENV)</td>
</tr>
<tr>
<td><strong>DSNTIPR1</strong></td>
<td>The following fields:</td>
</tr>
<tr>
<td></td>
<td>All fields (WLMENVx).</td>
</tr>
<tr>
<td><strong>DSNTIPRA</strong></td>
<td>On panels DSNTIPRB - DSNTIPRQ:</td>
</tr>
<tr>
<td></td>
<td>All WLM ENV NAME fields</td>
</tr>
<tr>
<td></td>
<td>(RTxvinWLM)</td>
</tr>
<tr>
<td><strong>DSNTIPY</strong></td>
<td>All numbered job card information fields.</td>
</tr>
</tbody>
</table>

2. Create the workflow definition files and input variable files by running the installation CLIST and specifying USE Z/OSMF WORKFLOW === YES on panel.
DSNTIPA1. When you specify that option, the CLIST runs with modified and extra panels that support generation of the z/OSMF workflow artifacts.

**Tip:** You can specify an HFS or zFS directory for storing the z/OSMF artifacts. Specify the directory in the PATH FOR Z/OSMF ARTIFACTS field on panel DSNTIPM1.

For more information about the specific panels that support the generation of z/OSMF workflow artifacts, see "Installation panels for installing or migrating with z/OSMF" on page 579.

a. Create the workflow definition file for migrating the first subsystem to Db2 12 by running the installation CLIST and specifying the following field values on panel DSNTIPA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> MIGRATE
- DATA SHARING ===> YES, with option 1. Group selected on DSNTIP0A

As you complete the CLIST, select each subsystem parameter that you want to include in the input variable file. When you complete the CLIST, the DSNTIWMS and DSNTIWMMN workflow definition files and associated input variable files are created.

b. If you are migrating a data sharing group, create the workflow definition file for migrating other data sharing members (DSNITWMD) by running the CLIST again and setting the following options on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> MIGRATE
- DATA SHARING ===> YES, with option 2. Member selected on DSNTIP0A

As you complete the CLIST, select each subsystem parameter that you want to include in the input variable file. When you complete the CLIST, the DSNITWMD workflow definition file and the DSNTIVMD input variable file are created.

3. Create extra input variable files, as required to support the efficient migration of your environment. The goal is to structure the input variable files so that you can specify subsystem parameters that are common across many subsystems in a single file. For an example of how you might and structure and name a set of input variable files for migrating a particular environment, see "Example of migrating a data sharing group to Db2 12 by using z/OSMF" on page 578.

**Important:** All input variable files must include the following variables, with supported values:

**INSTYPE**

The installation type. The following values are accepted:

- **ACTV**
  - Activating a Db2 12 function level.

- **MNDS**
  - Migration of a non-data-sharing subsystem to Db2 12.

- **MDS1**
  - Migration of the first member of a data sharing group to Db2 12.

- **MDSS**
  - Migration of subsequent members of a data sharing group to Db2 12.
MNFM
Activation of Db2 12 new function. (Function level 500 or higher.)
INDS Installation of a non-data-sharing Db2 subsystem.
IDS1 Installation of the first member of a data sharing group
IDSS Adding a member to a data sharing group

NEWSAMP2=
NEWSAMP2 is the partitioned data that contains tailored JCL templates
for the z/OSMF workflow. You must change this value if the workflow
artifacts are moved to a different location.
B= The indicator for line breaks in z/OSMF. Do not change or remove this
variable in a generated input variable file.

4. Create the workflow instances in z/OSMF.
   a. Specify the path of the workflow definition file and the input variable file.
      The path of the input variable file is optional, but it is best to always
      specify it. Otherwise, z/OSMF prompts you to provide the variable values
during execution, preventing full automation of the process.
   b. Specify a name for the workflow instance.
      Tip: Specify a name that describes the purpose of the workflow and
      includes the name of the target system.
   c. Assign all workflow steps to the same person to fully automate the
      migration process.
   d. Validate the JCL job and the substituted input variables. You can find the
      input variable files and the JCL jobs with the substituted input variables on
      the Perform tab for the workflow in z/OSMF.
      1) Review the input variables. You can edit the values of input variables
during your review, but any changes that you make apply to this
workflow instance only.
      2) In the Create JOB statement step, z/OSMF prompts you to review and
      update the JOB statement JCL. You can specify appropriate CLASS or
      MSGCLASS values.
      If installation or migration step requires authorization, you can also
      specify the USERNAME and PASSWORD. For detailed information
      about required authorizations see Required authorization for installation
      or migration
      3) Review the JCL including the JOB statement. You can click Edit JCL to
      make changes. However, any changes that you make apply only to this
      workflow instance. It is best to make any changes that apply to all
      instances in the JCL template before you create the instances.

5. Start running the workflow steps. You can run the workflow steps in z/OSMF,
or by embedding z/OSMF REST services in your own automation programs.
   • For a single Db2 subsystem, run the migration steps in the following order:
      a. Run DSNTIWMS to migrate to subsystem to Db2 12.
      b. Run DSNTIWMN to activate Db2 12 new function (function level 500 or
      higher). Run DSNTIWMN to migrate the subsystem to Db2 12
      new-function mode.
   • For a data sharing group, run the migration steps in the following order:
      a. Run DSNTIWMS to migrate the first member to Db2 12.
b. Run DSNTIWMD to migrate each subsequent member to Db2 12.

c. Run DSNTIWMN to migrate the data sharing group to activate Db2 12
   new function (function level 500 or higher).

a. Certain workflow steps, such as the pre-migration job DSNTIJPM, might
   need to be run repeatedly. For such repeated steps, you can select **Manually
   perform from the selected step.** After you run the step, it is marked
   “Completed” in z/OSMF. However, you can still rerun this step. That is,
   you can take the required actions that are identified in the premigration
   reports, and then return to z/OSMF to repeat this step until all
   premigration tasks are completed.

b. After any manual steps are complete, select **Automatically perform the
   selected step, and all subsequent steps...** on the Perform Automated Step
   page. z/OSMF runs all subsequent steps in order until all steps complete
   successfully, or until an error is encountered.

c. If errors are encountered, check the job output on the z/OSMF Status tab.
   The maximum return code that a step tolerates is defined the workload
   definition file, and it might be a return code other than 0. The maximum
   return code for most steps is 0 or 4.

**Example of migrating a data sharing group to Db2 12 by using
z/OSMF**

Assume that a data sharing group contains members, 12 members that for online
transaction processing (OLTP) workloads, and 4 members for online analytics
processing (OLAP) workloads.

- The group consists of 16 members named like DBxA, where x is numbers 1 - 9
  and letters A - G.
- Twelve members, DB1A - DBCA, process OLTP workloads.
- Four members, DBDA - DBGA, process OLAP workloads.
- For migration to Db2 12 function level 100, the following artifacts are used:
  - Member DB1A uses the DSNTIWMS workflow.
  - All other members use the DSNTIWMD workflow.
  - Members that process OLTP workloads share the DSNTIVTP input variable
    file.
  - Members that process OLAP workloads share the DSNTIVAP input variable
    file.
  - Every member also has an input variable file for member-specific parameters,
    each named like DSNTIVxA, where x corresponds to the matching character
    in the subsystem name.
- For the activation of function level 500 or higher, a single members uses the
  DSNTIWMN workflow and the DSNTVNTP input variable file.

The following table shows the members of the example data sharing group and the
workflow artifacts that can be used to migrate each subsystem.
Table 51. Workflow artifacts for migrating an example 16-member data sharing group

<table>
<thead>
<tr>
<th>Subsystem use</th>
<th>Subsystem name</th>
<th>Workflow files</th>
<th>Variables files</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLTP</td>
<td>DB1A</td>
<td>DSNTIWMS and DSNTIWMN</td>
<td>DSNTIV1A and DSNTIVNTP</td>
</tr>
<tr>
<td></td>
<td>DB2A</td>
<td></td>
<td>DSNTIV2A</td>
</tr>
<tr>
<td></td>
<td>DB3A</td>
<td></td>
<td>DSNTIV3A</td>
</tr>
<tr>
<td></td>
<td>DB4A</td>
<td></td>
<td>DSNTIV4A</td>
</tr>
<tr>
<td></td>
<td>DB5A</td>
<td></td>
<td>DSNTIV5A</td>
</tr>
<tr>
<td></td>
<td>DB6A</td>
<td></td>
<td>DSNTIV6A</td>
</tr>
<tr>
<td></td>
<td>DB7A</td>
<td></td>
<td>DSNTIV7A</td>
</tr>
<tr>
<td></td>
<td>DB8A</td>
<td>DSNTIWMD</td>
<td>DSNTIV8A</td>
</tr>
<tr>
<td></td>
<td>DB9A</td>
<td></td>
<td>DSNTIV9A</td>
</tr>
<tr>
<td></td>
<td>DBAA</td>
<td></td>
<td>DSNTIVAA</td>
</tr>
<tr>
<td></td>
<td>DBBA</td>
<td></td>
<td>DSNTIVBA</td>
</tr>
<tr>
<td></td>
<td>DBCA</td>
<td></td>
<td>DSNTIVCA</td>
</tr>
<tr>
<td>OLAP</td>
<td>DBDA</td>
<td></td>
<td>DSNTIVDA</td>
</tr>
<tr>
<td></td>
<td>DBEA</td>
<td></td>
<td>DSNTIVEA</td>
</tr>
<tr>
<td></td>
<td>DBFA</td>
<td></td>
<td>DSNTIVFA</td>
</tr>
<tr>
<td></td>
<td>DBGA</td>
<td></td>
<td>DSNTIVGA</td>
</tr>
</tbody>
</table>

What to do next

Activate function level 500 or higher, as described in [Activating Db2 function levels](Activating Db2 function levels) by using z/OSMF (Db2 for z/OS What’s New?).

Related concepts:
- [About workflows (z/OS Management Facility)](About workflows (z/OS Management Facility))

Related tasks:
- [Migrating Db2 to Db2 12](Migrating Db2 to Db2 12)

Related reference:
- [DSNTIPA1 or DSNTILA1: Main panel](DSNTIPA1 or DSNTILA1: Main panel)
- [IBM z/OS Management Facility Online Help](IBM z/OS Management Facility Online Help)
- [z/OSMF workflow: Migrate data sharing members to Db2 12 function level 100: DSNTIWMD and DSNTIVMD](z/OSMF workflow: Migrate data sharing members to Db2 12 function level 100: DSNTIWMD and DSNTIVMD)

Related information:

Installation panels for installing or migrating with z/OSMF

If you specify `USE Z/OSMF WORKFLOW === YES` on panel DSNTIPA1, panel DSNTILA1 opens, and the CLIST uses a modified set of installation panels.

Panel DSNTILA1

This panel opens to replace panel DSNTIPA1 when you specify `USE Z/OSMF WORKFLOW === YES`. It is a variant of DSNTIPA1, but you can select certain fields to tailor the values as variables in the z/OSMF input variable files. For more...
When you run the CLIST to prepare z/OSMF workflow definition files, the CLIST panels are modified so that you can select fields to tailor the values as variables in the z/OSMF input variable file.

You can find the variable names for each panel field and subsystem parameter in Directory of subsystem parameters, panel fields, and application default values.

For example, the following figure shows the DSNTIPA2 panel. You can select each of the panel fields except for DEFINE CATALOG, which is not eligible to be selected as an input variable. In this example, the CATALOG ALIAS field is selected as an input variable.

![Figure 81. The Install Db2 - z/OSMF artifacts configuration panel: DSNTIPM1](image)

**WORKFLOW DEFINITION LIBRARY**

The data set that stores the generated workflow definition files and input variable files.
JCL TEMPLATE LIBRARY
The data set that stores the tailored jobs for the z/OSMF workflow.

PATH FOR Z/OSMF ARTIFACTS
The name of the HFS or zFS path where CLIST processing places the
generated z/OSMF variable file, workflow file, and JCL templates. The
default setting of blank indicates that the installation CLIST does not copy
generated z/OSMF artifacts into HFS or zFS. When the field is not blank,
the entry is validated to ensure that the specified path does not already
exist as a file, and the user ID running the CLIST is authorized to create
the specified directory and to write content into it.

PRIOR RELEASE SDSNLOAD LIBRARY
For migration only, the location of the SDSNLOAD library from the
previous Db2 release. The field is blank because the CLIST cannot
determine the name of this library. You must specify a value for this field.

PRIOR RELEASE SDSNEXIT LIBRARY
For migration of a Db2 subsystem or the first or subsequent members of a
data sharing group, the name of the Db2 SDSNEXIT library from the
previous Db2 release. This field cannot be blank. Its initial value is set as
follows:
• The OLDEXIT setting from the current release DSNTIDXA/XB input
  parameter, if it is not blank.
• Otherwise, the EXIT LIBRARY field on panel DSNTIPT, which is the
  current release Db2 SDSNEXIT library.

You can modify this field and select this field to tailor its value as a
variable in the z/OSMF variable input file. The PRIOR RELEASE
SDSNEXIT LIBRARY field is only relevant when migrating a Db2
subsystem or the first or subsequent members of a data sharing group.
Otherwise, this field is set to blank. You cannot modify the setting nor can
you select it to tailor its value as a variable in the z/OSMF variable input
file.

PRIOR RELEASE PARAMETER MODULE
For migration of a Db2 subsystem or the first member of a data sharing
group only, the name of the Db2 subsystem parameter module from the
previous Db2 release. This field cannot be blank.

The initial values is set based on the following settings:
1. The OLDZPARM setting from the current release DSNTIDXA or
   DSNTIDXB input parameter, unless it is blank.
2. The PARAMETER MODULE field on panel DSNTIP03, which is the
   Db2 subsystem parameter module from the current release.

You can modify this field and select this field to tailor its value as a
variable in the z/OSMF variable input file.

Panels DSNTIPM3 and DSNTIPM4
Contain fields for the subsystem parameters that do correspond to fields on any
DSNTIPxx panels. For descriptions of the parameters on these panels, see
Subsystem parameters that are not on installation panels

You can select each panel field to specify that the value is tailored as a variable
and stored in the z/OSMF input variable file.
Panel DSNTIPM5

Contains parameters to tailor the following installation jobs:
- DSNTIJPM
- DSNTIJSA
- DSNTIJSO

Like the DSNTIPxx panels, you can select each panel field to specify that the value is tailored as a variable that is stored in the z/OSMF input variable file.

Panels for choosing z/OSMF workflow steps

On the following scenario-specific panels you can choose the steps to include in z/OSMF workflows for completing Db2 12 installation or migration tasks.

Each such panel indicates the required and optional steps, and whether they are selected by default, for example:
- S indicates a required step
- $ indicates an optional step that is selected by default
- _ indicates an optional step that is not selected by default

Panel DSNTIPMS

Panel DSNTIPMS is a scenario-specific panel for choosing steps for migrating a non-data sharing Db2 subsystem or the first member of a data sharing group to Db2 12.
Notes:
1. Complete this step before migration.
2. If this job is selected, job DSNTIJISO runs before to stop Db2, and job DSNTIJSB runs after to start Db2.
3. DSNTIJJIB is tailored based on DSNTIJIC, using the Db2 11 SDSNLOAD library.

The steps on panel DSNTIPMS continue on panel DSNTIPMT:

DSNTIPMT REVIEW STEPS IN Z/OSMF MIGRATION WORKFLOW DEFINITION (PANEL 2)

Select additional steps which should be included in the workflow definition:

- Start DB2 Version 12 (DSNTIJSA)
- Tailor DB2 Version 12 catalog (DSNTIJTC)
- Ensure that the catalog indexes have no problems (DSNTIJCX)
- Prepare dynamic SQL program (DSNTIJTM)
- Bind SPUFI and DCLGEN and user-maintained DB activity (DSNTIJSG)
- Set up DB2-supplied routines (DSNTIJRW)
- Configure DB2 for running SPs and user-defined routines (DSNTIJRT)
- Validate the installation of DB2-supplied routines (DSNTIJRV)
- Take an image copy of the DB2 catalog (DSNTIJIC)

PRESS: ENTER to continue  RETURN to exit  HELP for more information

For more information about the steps, see [Migrating your Db2 subsystem to Db2 12](#).

Panels DSNTIPMS and DSNTIPMT are shown only when you specify These panels are shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> MIGRATE
- DATA SHARING ===> NO, or DATA SHARING ===> YES with option 1 selected in the DSNTIPOB window.

Panel DSNTIPMN

Panel DSNTIPMN scenario-specific panel for choosing steps for completing the Db2 12 migration process.
For more information about the steps, see [Migrating your Db2 subsystem to Db2 12](#), beginning with [Migration step 27: Take an image copy of the Db2 12 catalog: DSNTIJIC](#).

DSNTIPMN is shown only when you specify the following values on panel DSNTILA1:

- **USE Z/OSMF WORKFLOW**: YES
- **INSTALL TYPE**: MIGRATE
- **DATA SHARING**: NO, or **DATA SHARING**: YES with option 1 selected in the DSNTIP0B window.

**Panel DSNTIPMD**

A scenario-specific panel for choosing steps for migrating subsequent members of a data sharing group members to Db2 12.

For more information about the steps, see [Migrating subsequent members of a group to Db2 12](#). The workflow supports only steps that are shown in the panel.

DSNTIPMD is shown only when you specify the following values on panel DSNTILA1:

- **USE Z/OSMF WORKFLOW**: YES
- **INSTALL TYPE**: MIGRATE
DATA SHARING ===> YES with option 2 selected in the DSNTIPOB window.

Panel DSNTIPMI

Panel DSNTIPMI is a scenario-specific panel for choosing steps to include when installing a new Db2 subsystem.

<table>
<thead>
<tr>
<th>REVIEW STEPS IN Z/OSMF INSTALLATION WORKFLOW DEFINITION (PANEL 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIPMI REVIEW STEPS IN Z/OSMF INSTALLATION WORKFLOW DEFINITION (PANEL 1)</td>
</tr>
<tr>
<td>====&gt;</td>
</tr>
<tr>
<td>Select the steps which should be included in the workflow definition:</td>
</tr>
<tr>
<td>s Define DB2 to z/OS (DSNTIJMV)</td>
</tr>
<tr>
<td>s Add DB2 for z/OS address space startup JCL procedures (DSNTIJMA)</td>
</tr>
<tr>
<td>s Add DB2 program preparation and utilities invocation JCL proc (DSNTI JMP)</td>
</tr>
<tr>
<td>s Add address space JCL procs for the DB2-supplied WLM environments (DSNTIJMW)</td>
</tr>
<tr>
<td>s Add environment files for the DB2-supplied Java WLM environment (DSNTIJMJ)</td>
</tr>
<tr>
<td>s Define the ICF catalog and alias (DSNTIJCA)</td>
</tr>
<tr>
<td>s Define system data sets (DSNTIJIN)</td>
</tr>
<tr>
<td>s Define the DB2 subsystem parameter module (DSNTIJZ)</td>
</tr>
<tr>
<td>s Define the DB2 application defaults module (DSNTIJUA)</td>
</tr>
<tr>
<td>s Define the DB2 offline message CCSIDs module (DSNTIJUM)</td>
</tr>
<tr>
<td>s Update DDF related information in the BSOS (DSNTIJUL)</td>
</tr>
<tr>
<td>s Initialize system data sets - 1 (DSNTIJID)</td>
</tr>
<tr>
<td>s Initialize system data sets - 2 (DSNTIJIE)</td>
</tr>
<tr>
<td>s Initialize system data sets - 3 (DSNTIJIF)</td>
</tr>
<tr>
<td>s Define user authorization exit routines (DSNTIJEX)</td>
</tr>
<tr>
<td>s Connect DB2 to TSO (DSNTIJVC)</td>
</tr>
</tbody>
</table>

PRESS: ENTER to continue  RETURN to exit  HELP for more information

The steps on panel DSNTIJPMI continue on panel DSNTIPMJ.

<table>
<thead>
<tr>
<th>REVIEW STEPS IN Z/OSMF INSTALLATION WORKFLOW DEFINITION (PANEL 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIPMJ REVIEW STEPS IN Z/OSMF INSTALLATION WORKFLOW DEFINITION (PANEL 2)</td>
</tr>
<tr>
<td>====&gt;</td>
</tr>
<tr>
<td>Select the steps which should be included in the workflow definition:</td>
</tr>
<tr>
<td>s Start the DB2 subsystem (DSNTIJSA)</td>
</tr>
<tr>
<td>s Tailor the DB2 catalog (DSNTIJTC)</td>
</tr>
<tr>
<td>s Activate a DB2 function level (DSNTIJAF)</td>
</tr>
<tr>
<td>s Create default storage group, define temporary work files, and bind DB2 REXX Language Support (DSNTIJM)</td>
</tr>
<tr>
<td>s Define and bind DB2 objects (DSNTIJSG)</td>
</tr>
<tr>
<td>s Setting up the DB2 core WLM environments during migration (DSNTIJRW)</td>
</tr>
<tr>
<td>s Installing DB2-supplied routines during migration (DSNTIJRT)</td>
</tr>
<tr>
<td>s Validate the installation of DB2-supplied routines (DSNTIJRV)</td>
</tr>
<tr>
<td>s Back up the DB2 directory and catalog (DSNTIJIC)</td>
</tr>
</tbody>
</table>

PRESS: ENTER to continue  RETURN to exit  HELP for more information

For more information about the steps, see Installing Db2.

DSNTIPMI is shown only when you specify the following values on panel DSNTILA1:

| USE Z/OSMF WORKFLOW ===> YES |
| INSTALL TYPE ===> INSTALL |
| DATA SHARING ===> NO, or DATA SHARING ===> YES with option 1 selected in the DSNTIPOA window.

Panel DSNTIPMA

Panel DSNTIPMA is scenario-specific panel for choosing steps for adding a Db2 subsystem to a data sharing group.
DSNTIPMA REVIEW STEPS IN Z/OSMF INSTALLATION WORKFLOW DEFINITION

Select the steps which should be included in the workflow definition:

- Define DB2 to z/OS (DSNHIJM)
- Add DB2 address space startup JCL procedures (DSNHIJMA)
- Define system data sets (DSNHIJIN)
- Define the DB2 subsystem parameter module (DSNHIJUZ)
- Update DDF related information in the BSDS (DSNHIJUL)
- Initialize system data sets (DSNHIJID/DSNHIJIE/DSNHIJIF)
- Define user authorization exit routines (DSNHIJEX)
- Start the DB2 subsystem (DSNHIJSA)
- Initialize system data sets (DSNHIJID/DSNHIJIE/DSNHIJIF)

Press: ENTER to continue RETURN to exit HELP for more information

For more information about the steps, see Adding a new data sharing member. The workflow supports only steps that are shown in the panel.

DSNTIPMA is shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> INSTALL
- DATA SHARING ===> YES with option 2 selected in the DSNTIPOA window.

Panel DSNTIPMV

Panel DSNTIPMV is a scenario-specific panel for choosing steps to include when running installation verification jobs.

Select the IVP steps which should be included in the workflow definition:

- Remove sample applications and sample schema authorizations (DSNTEJ0)
- Create and load sample tables (DSNTEJ1)
- Prepare and invoke program DSNTEP2 (DSNTEJ1L)
- Prepare and invoke program DSNTEAUL (DSNTEJ2A)

Press: ENTER to continue RETURN to exit HELP for more information

For more information about the steps, see Verifying successful Db2 12 installation or migration with sample applications.

DSNTIPMV is shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> INSTALL or MIGRATE
- DATA SHARING ===> YES

Panel DSNTIPMF

Panel DSNTIPMF is a scenario-specific panel for choosing steps to include when activating a Db2 function level.
For more information about the steps, see Migration step 29: Activate function level 500.

This panel is shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> ACTIVATE
- DATA SHARING ===>

DSNTIPMU

Contains fields for specifying source and target workflow input variable files, for use when creating input variable files for installing or migrating members of a data sharing group after the first one.

This panel is shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> UPDATE

Related concepts:
- About workflows (z/OS Management Facility)
- Sample workflow artifacts for installation and migration with z/OS Management Facility

Sample jobs for installing and migrating with z/OSMF

The following sample jobs are used by z/OSMF workflows for automating the installation or migration of Db2 subsystems.

**Job DSNTIJDC**

The DSNTIJDC sample job creates image copies of the Db2 catalog and directory. It is used by z/OSMF workflows for automating the installation or migration of Db2 subsystems.

**Job DSNTIJISO**

The DSNTIJISO sample job stops Db2 by calling the DSNTMVSBU stand-alone program. It is used by z/OSMF workflows for automating the installation or migration of Db2 subsystems.
Job DSNTIJSA

The DSNTIJSA sample job starts Db2 by calling the DSNTMVSB stand-alone program. It is used by z/OSMF workflows for automating the installation or migration of Db2 subsystems.

Job DSNTIJMB

Job DSNTIJMB runs the DSNTMVSB batch program tool to issue START DB2 and STOP DB2 commands.

Related tasks:

Automating Db2 installation or migration with z/OS Management Facility

Related reference:

DSNTMVSB tool

DD statements for DSNTMVSB

The DSNTMVSB tool provides a batch interface for starting and stopping Db2. This tool is invoked by sample jobs DSNTIJSA and DSNTIJSO to support z/OSMF workflows for automating installation and migration of Db2 subsystems.

Input DD statements for DSNTMVSB

CMDOPT

For command and command parameters used to generate the command to be executed:

TYPE

Specifies the type of command: Db2

Required.

COMMAND PREFIX

Specifies the command prefix that identifies which Db2 subsystem to direct the command to.

Required.

REMOTE SYSTEM

Specifies the z/OS system in the sysplex to which the Db2 command is to be routed.

Required if the Db2 command will be routed to a z/OS system that is different from the system where the batch program DSNTMVSB is executing.

COMMAND

Specifies the command to be executed: START or STOP

Required.

PARAMETERS

Specifies the parameters of the command. START DB2 options, such as PARM(DB2AZNS), if COMMAND=START
STOP DB2 options, such as MODE(FORCE), if COMMAND=STOP
Optional.

**SUBPARAMETERS**

Specifies the subparameters of the command, represented as keyword-value pairs separated by commas:

**CONSNAMES=consame**
Where **consame** is the name of the console used to issue the command. If not specified, the default is DSNTMVSBB. The specified console must not already be active.
Optional.

**IRLMPROC=irlmproc**
Where **irlmproc** is the procedure name of the connected IRLM. If specified, the batch program will poll the status of the specified IRLM job and will report the end of Db2 normal termination when the IRLM job has ended within the specified wait_timeout.
Optional. Valid if COMMAND=STOP and SSNM keyword is not specified.

**SSNM=sqnsm**
Where **sqnsm** is the Db2 subsystem name. If specified, the batch program will poll the status of the specified Db2 **sqnsm**MSTR job and will report the end of Db2 normal termination when the **sqnsm**MSTR job has ended within the specified wait_timeout.
Optional. Valid if COMMAND=STOP and IRLMPROC keyword is not specified.

**WAIT_TIMEOUT**
Specifies the amount of time in seconds that this batch program waits for the command to complete and for the command messages to be routed to the console. It includes time spent waiting for the console to be available and polling the **irlmproc** or **sqnsm**MSTR job status.

1 -900. The default value is 180 seconds.

**Output DD statements used by DSNTIMVSB**

**SYSPRINT**
For DSNTIMVSB messages.

**CMDOUT**
For command messages.

**DSNTRACE**
For trace messages.
DSNTMVS return codes

DSNTMVS sets the following return-code and command-completion-code variables to describe the execution status of the command:

return-code

0

The batch program did not encounter any errors during processing.

command-completion-code is set to 0 or 4.

4

The batch program issued the command but was unable, within the prescribed wait time, to determine the command execution status based on the command messages retrieved. All command messages and any Db2 MSTR / IRLM job status messages retrieved within the prescribed wait time are written to DD:CMDOU.

command-completion-code is set to 8 or 16.

8

The batch program issued the command but was unable to determine the command execution status due to an internal error. All the command messages and any Db2 MSTR / IRLM job status messages retrieved so far are written to DD:CMDOU.

command-completion-code is set to 8 or 16.

12

The batch program encountered an error during processing. A message describing the error is written to SYSPRINT.

16

The batch program did not encounter any errors during processing. However, the command completed abnormally.

command-completion-code is set to 12.

command-completion-code

0

The command completed successfully.

Table 52. Details for command-completion-code 0

<table>
<thead>
<tr>
<th>Command</th>
<th>command-completion-code meaning</th>
<th>Message received</th>
</tr>
</thead>
<tbody>
<tr>
<td>START DB2</td>
<td>Successful completion</td>
<td>DSN9022I cmd_prefix</td>
</tr>
<tr>
<td>STOP DB2</td>
<td>Successful completion</td>
<td>DSN9022I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 53. Details for command-completion-code 4

<table>
<thead>
<tr>
<th>Command</th>
<th>command-completion-code meaning</th>
<th>Message received</th>
</tr>
</thead>
<tbody>
<tr>
<td>START DB2</td>
<td>Db2 was already active when the</td>
<td>DSNY003I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td>command was issued.</td>
<td></td>
</tr>
</tbody>
</table>
Table 53. Details for command-completion-code 4 (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>command-completion-code meaning</th>
<th>Message received</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP DB2</td>
<td>When the command was issued, Db2 was already stopped or Db2 was in the process of stopping and it stopped successfully.</td>
<td>If Db2 MSTR or IRLM address space termination monitoring is not requested, any of the following messages are received:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DSN3106I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DSNY004I cmd_prefix and DSN9022I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If Db2 MSTR address space termination monitoring is requested, both of the following messages were received:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DSNY004I cmd_prefix and DSN9022I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 'ssnm MSTRNOT FOUND' in 'D J, ssnm MSTR' command output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If IRLM address space termination monitoring is requested, both of the following messages were received:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DSNY004I cmd_prefix and DSN9022I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 'irlm_job NOT FOUND' in 'D J, 'irlm_job' command output</td>
</tr>
</tbody>
</table>

8

The command started but the completion status is unknown.

Table 54. Details for command-completion-code 8

<table>
<thead>
<tr>
<th>Command</th>
<th>command-completion-code meaning</th>
<th>Message received</th>
<th>Expected message not received</th>
</tr>
</thead>
<tbody>
<tr>
<td>START DB2</td>
<td>Db2 is in the process of starting, but the completion status of the START DB2 command is unknown.</td>
<td>DSNY024I cmd_prefix</td>
<td>Any of the following messages:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSN9023I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSNV086E cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSN3104I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSN9022I cmd_prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• DSN3100I cmd_prefix</td>
</tr>
</tbody>
</table>
Table 54. Details for command-completion-code 8 (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>command-completion-code meaning</th>
<th>Message received</th>
<th>Expected message not received</th>
</tr>
</thead>
</table>
| STOP DB2 | Db2 is in the process of stopping, but the completion status of the STOP DB2 command is unknown. | One of the following messages was received:  
  - DSNY002I cmd_prefix  
  - DSNY004I cmd_prefix | One of the following messages was not received:  
  - DSN9022I cmd_prefix  
  - DSN9023I cmd_prefix  
  Or DSN9022I cmd_prefix was received along with one of the following conditions:  
  - Db2 MSTR address space termination monitoring is requested, and 'ssnmMSTR NOT FOUND' was not found in 'D J, ssnmMSTR' command output  
  - IRLM address space termination monitoring is requested, and 'irlm_job NOT FOUND' is not found in 'D J, irlm_job' command output. |

12

The command completed abnormally.

Table 55. Details for command-completion-code 16

<table>
<thead>
<tr>
<th>Command</th>
<th>command-completion-code meaning</th>
<th>Message received</th>
</tr>
</thead>
</table>
| START DB2 | Abnormal completion. | One of the following messages was received:  
  - DSN9023I cmd_prefix  
  - DSNV086E cmd_prefix  
  - DSN3100I cmd_prefix  
  - DSN3104I cmd_prefix  
  - DSN3106I cmd_prefix  
  - DSN3107I cmd_prefix |
| STOP DB2  | Abnormal completion. | One of the following messages was received:  
  - DSN9023I cmd_prefix  
  - DSN3107I cmd_prefix |

16

The command output does not satisfy the conditions that are listed for command-completion-code 0, 4, 8, or 12.

Related tasks:
- [Automating Db2 installation or migration with z/OS Management Facility](#)

Related reference:
- [Job DSNTIJMB](#)
- [START DB2 (Db2) (Db2 Commands)](#)
Job DSNTIJ0Z: bring Db2 subsystem parameter changes online

Job DSNTIJ0Z runs a SET SYSPARM command to bring subsystem parameter changes online.

The z/OSMF workflows for migrating to Db2 12 and activating Db2 function levels use this job.

Related tasks:

Automating Db2 migration by using z/OS Management Facility

Related reference:

Sample workflow artifacts for installation and migration with z/OS Management Facility

Db2 provides sample z/OSMF workflow artifacts for Db2 installation and migration. The artifacts include several sample workflow definition files, and an associated a set of sample JCL templates and a sample workflow input variable file for each definition file.

You can find the sample z/OSMF workflow artifacts in the prefix.SDSNIVPD Db2 SMP/E library.

You can find the steps of each workflow on one or more scenario-specific panels. Each workflow supports only the steps that are shown on the panel.

Required, optional, and default workflow steps

Each panel indicates the required and optional steps and whether they are selected by default, for example:

- Indicates a required step
- Indicates an optional step that is selected by default
- Indicates an optional step that is not selected by default

DSNTIWMS and DSNTIVMS: migrate to Db2 12

The DSNTIWMS workflow migrates a non-data sharing Db2 subsystem, or the first Db2 member of a data sharing group, to Db2 12. It uses the DSNTIVMS input variable file.

Panel DSNTIPMS shows the steps of the DSNTIWMS workflow.
DSNTIPMS REVIEW STEPS IN Z/OSMF MIGRATION WORKFLOW DEFINITION (PANEL 1)

Select the steps which should be included in the workflow definition:

- Actions to complete before migration (DSNTIJPM)
  - Convert BSDS records to support 10-byte RBA and LRSN fields (DSNTIJCB)
  - Take image copies of the directory and catalog (DSNTIJIB)
  - Make DB2 CLISTS available to TSO and batch users (DSNTIJVC)
  - Stop DB2 Version 11 activity (DSNTIJSO)
  - Define the DB2 subsystem parameter module (DSNTIJUZ)
  - Define the DB2 application defaults module (DSNTIJUA)
  - Define the DB2 offline message CCSIDs module (DSNTIJUM)
  - Define DB2 12 to z/OS (DSNTIJMV)
  - Add DB2 program preparation and utilities invocation JCL proc (DSNTIJMP)
  - Add address space JCL procs for the DB2-supplied WLM environments (DSNTIJMU)
  - Add environment files for the DB2-supplied Java WLM environment (DSNTIJMU)
  - Take an image copy of the DB2 catalog (DSNTIJIC)

PRESS: ENTER to continue RETURN to exit HELP for more information

Notes:
1. Complete this step before migration.
2. If this job is selected, job DSNTIJSO runs before to stop Db2, and job DSNTIJSB runs after to start Db2.
3. DSNTIJIB is tailored based on DSNTIJIC, using the Db2 11 SDSNLOAD library.

The steps on panel DSNTIPMS continue on panel DSNTIPMT:

DSNTIPMT REVIEW STEPS IN Z/OSMF MIGRATION WORKFLOW DEFINITION (PANEL 2)

Select additional steps which should be included in the workflow definition:

- Start DB2 Version 12 (DSNTIJSA)
  - Tailor DB2 Version 12 catalog (DSNTIJTC)
    - Ensure that the catalog indexes have no problems (DSNTIJCX)
  - Prepare dynamic SQL program (DSNTIJTM)
  - Bind SPUFI and DCLGEN and user-maintained DB activity (DSNTIJSG)
  - Set up DB2-supplied routines (DSNTIJRW)
  - Configure DB2 for running SPs and user-defined routines (DSNTIJRT)
  - Validate the installation of DB2-supplied routines (DSNTIJRV)
  - Take an image copy of the DB2 catalog (DSNTIJIC)

PRESS: ENTER to continue RETURN to exit HELP for more information

For more information about the steps, see [Migrating your Db2 subsystem to Db2](#)

Panels DSNTIPMS and DSNTIPMT are shown only when you specify These panels are shown only when you specify the following values on panel DSNTILAA:

USE Z/OSMF WORKFLOW ===> YES
INSTALL TYPE ===> MIGRATE
DATA SHARING ===> NO, or DATA SHARING ===> YES with option 1 selected in the DSNTIP0B window.
DSNTIWMD and DSNTIVMD: migrate data sharing members to Db2 12

The DSNTIWMD workflow migrates subsequent members of a Db2 data sharing group to Db2 12. It uses the DSNTIVMD input variable file.

Panel DSNTIPMD shows the steps of this workflow.

For more information about the steps, see Migrating subsequent members of a group to Db2 12. The workflow supports only steps that are shown in the panel.

DSNTIPMD is shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> MIGRATE
- DATA SHARING ===> YES with option 2 selected in the DSNTIP0B window.

DSNTIWMN and DSNTIVMN: activate Db2 12 function level 500 or higher

The DSNTIWMN sample workflow activates Db2 12 new function in the Db2 subsystem. It uses the DSNTIVMN input variable file.

Panel DSNTIPMN shows the steps of the DSNTIWMN workflow.

For more information about the steps, see Activate Db2 12 function level 500 or higher. The workflow supports only steps that are shown in the panel.
For more information about the steps, see Migrating your Db2 subsystem to Db2 12, beginning with Migration step 27: Take an image copy of the Db2 12 catalog: DSNTIJIC.

DSNTIPMN is shown only when you specify the following values on panel DSNTILA1:
- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> MIGRATE
- DATA SHARING ===> NO, or DATA SHARING ===> YES with option 1 selected in the DSNTIP0B window.

**DSNTIWAF and DSNTIVAF: activate a Db2 12 function level**

The DSNTIWAF workflow activates a specified Db2 12 function level in the Db2 subsystem. It can also be used to updated the default Db2 application compatibility (APPLCOMPAT subsystem parameter) setting and to bring that change online. It uses the DSNTIVAF input variable file.

Panel DSNTIPMF shows the steps of the DSNTIWAF workflow.

```
DSNTIPMF REVIEW STEPS IN Z/OSMF ACTIVATION WORKFLOW DEFINITION

Select the steps which should be included in the workflow definition:
- $ Back up the DB2 directory and catalog (DSNTIJIC)
- $ Tailor the DB2 catalog (DSNTIJTC)
- $ Activate a DB2 function level (DSNTIJAF)
- $ Rebuild the subsystem parameter module to change APPLCOMPAT level
- (DSNTIJUZ)
- $ Bring the APPLCOMPAT change online (DSNTIJQZ)
- $ Rebuild the application defaults module to change SQLLEVEL setting
- (DSNTIJUA)

PRESS: ENTER to continue RETURN to exit HELP for more information
```

For more information about the steps, see Migration step 29: Activate function level 500.

This panel is shown only when you specify the following values on panel DSNTILA1:
- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> ACTIVATE
- DATA SHARING ===>

**DSNTIWIN and DSNTIVIN: install Db2 12**

The DSNTIWIN sample workflow definition file installs a new Db2 12 subsystem.

Panel DSNTIPIPMI shows the steps of the DSNTIWIN workflow.
Select the steps which should be included in the workflow definition:
- Define DB2 to z/OS (DSNTIJMW)
- Add DB2 for z/OS address space startup JCL procedures (DSNTIJMA)
- Add DB2 program preparation and utilities invocation JCL procs (DSNTIJMP)
- Add address space JCL procs for the DB2-supplied WLM environments (DSNTIJMN)
- Add environment files for the DB2-supplied Java WLM environment (DSNTIJML)
- Define the ICF catalog and alias (DSNTIJCA)
- Define system data sets (DSNTIJIN)
- Define the DB2 subsystem parameter module (DSNTIJUZ)
- Define the DB2 system data sets (DSNTIJUZ)
- Define the DB2 subsystem parameter module (DSNTIJUZ)
- Define the DB2 application defaults module (DSNTIJUA)
- Add DB2 for z/OS address space startup JCL procedures (DSNTIJMA)
- Define the DB2 offline message CCSIDs module (DSNTIJUM)
- Update DDF related information in the BSDS (DSNTIJUL)
- Initialize system data sets - 1 (DSNTIJID)
- Initialize system data sets - 2 (DSNTIJIE)
- Initialize system data sets - 3 (DSNTIJIF)
- Define user authorization exit routines (DSNTIJEX)
- Connect DB2 to TSO (DSNTIJVC)

PRESS: ENTER to continue  RETURN to exit  HELP for more information

The steps on panel DSNTIPMI continue on panel DSNTIPMJ.

Select the steps which should be included in the workflow definition:
- Start the DB2 subsystem (DSNTIJSA)
- Tailor the DB2 catalog (DSNTIJTC)
- Create default storage group, define temporary work files, and bind DB2 REXX Language Support (DSNTIJTM)
- Define and bind DB2 objects (DSNTIJSG)
- Setting up the DB2 core WLM environments during migration (DSNTIJRW)
- Installing DB2-supplied routines during migration (DSNTIJRT)
- Validate the installation of DB2-supplied routines (DSNTIJRV)
- Back up the DB2 directory and catalog (DSNTIJIC)

PRESS: ENTER to continue  RETURN to exit  HELP for more information

For more information about the steps, see Installing Db2

DSNTIPMI is shown only when you specify the following values on panel DSNTILA1:

- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> INSTALL
- DATA SHARING ===> NO, or DATA SHARING ===> YES with option 1 selected in the DSNTIP0A window.

**DSNTIWIA and DSNTIVIA: add a new data sharing member**

The DSNTIWIA sample workflow definition file adds new members to a data sharing group. It uses the DSNTIVIA input variable file.

Panel DSNTIPMA shows the steps of the DSNTIWIA workflow.
Select the steps which should be included in the workflow definition:
- Define DB2 to z/OS (DSNTIJMV)
- Add DB2 address space startup JCL procedures (DSNTIJMA)
- Define system data sets (DSNTIJIN)
- Define the DB2 subsystem parameter module (DSNTIJUZ)
- Update DDF related information in the BSDS (DSNTIJUL)
- Initialize system data sets (DSNTIJID/DSNTIJIE/DSNTIJIF)
- Define user authorization exit routines (DSNTIJEX)
- Start the DB2 subsystem (DSNTIJS)
- Create default storage group, define temporary work files, and.
- bind DB2 REXX Language Support (DSNTIJTM)
- Back up the DB2 directory and catalog (DSNTIJIC)

For more information about the steps, see Adding a new data sharing member. The workflow supports only steps that are shown in the panel.

DSNTIPMA is shown only when you specify the following values on panel DSNTILA1:
- USE Z/OSMF WORKFLOW ===> YES
- INSTALL TYPE ===> INSTALL
- DATA SHARING ===> YES with option 2 selected in the DSNTIP0A window.

Related tasks:
- Automating Db2 installation by using z/OS Management Facility
- Automating Db2 migration by using z/OS Management Facility

598  Installation and Migration Guide
Chapter 8. Creating services for provisioning Db2 for z/OS subsystems (DBaaS)

You can use the sample Db2 software services template to create a service that rapidly provisions multiple standalone Db2 for z/OS subsystems, and deprovision them later.

Before you begin

Various administrators must provide information and prepare the z/OS environment for the Db2 software service template. For a list of these tasks, see “Preparing the environment for the Db2 software service template” in the readme file in GitHub.

About this task

The sample runs as a software service template in IBM Cloud Provisioning and Management for z/OS. For an overview, see IBM Cloud Provisioning and Management for z/OS.

You can use the sample Db2 software service template, to build your own Db2 software service template to provision multiple standalone Db2 12 for z/OS subsystem instances in a “typical Db2 configuration” based on the latest best practices. For details, see “About the sample Db2 software service template” in the readme file in GitHub.

The sample Db2 software service template also makes the following actions available in the service, for the provisioned Db2 subsystems:

- Start Db2
- Stop Db2
- Display DDF
- Enable optional features (Db2 REST services, ODBC, and JDBC)
- Deprovision the provisioned Db2 subsystem.

Tip: An alternative to using the sample Db2 software service template is to run the installation CLIST to generate the required artifacts. This approach supports more customization of the provisioned Db2 subsystems than the sample Db2 software service template. For more information, see Creating artifacts for provisioning Db2 subsystems by running the installation CLIST.

Procedure

To provision Db2 subsystems:

1. Download and prepare the sample template:
   a. Obtain the Db2ProvisionSystemNonDS.pax file. You can use the following approaches:
      - Locate the Db2ProvisionSystemNonDS.pax file installed in the directory specified by the DDDEF for SDSNASMP data set, and copy it to a directory in z/OS UNIX System Services.
b. Extract the `Db2ProvisionSystemNonDS.pax` file into the directory of your choice. For example:
```
pax -rvf Db2ProvisionSystemNonDS.pax
```

c. If you are using the sample artifacts before building your own template, you must edit the `dsntivin` input variable file, and update it according to your installation. For instructions see “Specifying input properties” in the readme file in GitHub.

2. Test the provisioning template, verify the provisioned Db2 subsystems, and test the deprovisioning process.

3. Publish the template to make it available to consumers.

Related information:

- [IBM Cloud Provisioning and Management for z/OS](#)
- [Cloud provisioning services](#)
Chapter 9. Installing, migrating, and enabling Db2 data sharing

Whether you need to install, migrate, or enable data sharing depends on the type of system that you start with and the type of system that you want to create.

Before you begin

Familiarize yourself with the following information:

- Planning for Db2 data sharing
- Strategy for creating a Db2 data sharing group
- Preparing for Db2 12 migration in Db2 data sharing

Procedure

To create or migrate a Db2 data sharing group:

Complete the appropriate task, as described in the following table.

<table>
<thead>
<tr>
<th>If you have this...</th>
<th>And you want this...</th>
<th>Do this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>No system</td>
<td>Db2 12 data sharing</td>
<td>Install a new data sharing group, as described in Installing a new data sharing group.</td>
</tr>
<tr>
<td>A Db2 12 non-sharing Db2 subsystem</td>
<td>The originating member of a data sharing group</td>
<td>Enable Db2 data sharing, as described in Enabling Db2 data sharing.</td>
</tr>
<tr>
<td>One member in the group</td>
<td>More members in the group</td>
<td>Add new data sharing members, as described in Adding a new data sharing member.</td>
</tr>
<tr>
<td>Separate Db2 subsystems</td>
<td>Merged Db2 subsystems into a single group</td>
<td>Merge existing Db2 data into the group, as described in Merging existing Db2 data into the group.</td>
</tr>
<tr>
<td>A Db2 11 data sharing group</td>
<td>A Db2 12 data sharing group</td>
<td>Migrate the existing data sharing group to Db2 12, as described in Migrating an existing data sharing group to Db2 12.</td>
</tr>
</tbody>
</table>

Related concepts:

Introduction to Db2 data sharing (Db2 Data Sharing Planning and Administration)

Related tasks:

Installing Db2
Migrating Db2 to Db2 12
Planning for Db2 data sharing

Db2 data sharing introduces special requirements for your Db2 and z/OS environments.

Related concepts:
- Release coexistence in Db2 data sharing

Related tasks:
- Preparing for Db2 12 migration in Db2 data sharing
- Installing, migrating, and enabling Db2 data sharing

Parallel Sysplex requirements for Db2 data sharing

Db2 data sharing depends on hardware and software components in the z/OS Parallel Sysplex.

Coupling facility

Install and define at least one coupling facility to z/OS before enabling the Db2 data sharing function. For high availability, more than one coupling facility is required.

Data sharing member names from the subsystem parameter load module (DSNZPARM) are used to connect members to the coupling facility at Db2 startup. The first connector causes the list structure to be allocated in a coupling facility based on the preference list in the active CFRM policy.

Related concepts:
- Subsystem parameters (Introduction to Db2 for z/OS)

Related tasks:
- Specifying a member’s subsystem parameter load module

Related reference:
- Adding MVS systems to a sysplex

Cross-system coupling facility component of z/OS

During startup, the members of a data sharing group join one cross-system coupling facility (XCF) group, and their associated internal resource lock managers (IRLMs) join another XCF group.

The z/OS cross-system extended services (XES) also join an XCF group implicitly on behalf of the IRLM connection to the lock structure. To join a particular group, the data sharing group members and the IRLMs use the names that you specify during Db2 installation.

Db2 uses the XCF for certain intersystem communications. Use both the coupling facility and channel-to-channel connections for XCF signaling. See Adding MVS systems to a sysplex for more information about configuring the XCF.

Coupling facility structures:

Db2 relies on areas of storage in the coupling facility called structures.
Three types of structures exist: lock, list, and cache. Each structure type has a unique function. The following figure shows a sample configuration of the coupling facility structures that are used by Db2.

![Coupling Facilities](image)

*Figure 82. Coupling facility structures that are used by Db2. This is a sample configuration. The lock structure and list structure (SCA) do not need to be in the same coupling facility.*

Members of a data sharing group use the following coupling facility structures to communicate and move data among themselves.

**Lock structure**

The lock structure protects shared Db2 resources (such as table spaces and pages) and enables concurrent access to those resources.

The system lock manager (SLM), a component of the z/OS cross-system extended services (XES), presents global lock information to the lock structure on behalf of each member’s IRLM.

The lock structure consists of two parts: a coupling facility lock list table (called the *modified resource list*), and a coupling facility lock hash table (called the *lock table*). The modified resource list records locks that protect changed data, thereby protecting the data in case of failure. The lock table contains the lock status information and the owning members of those locks, and is used to provide global lock serialization.

**List structure (SCA)**

The list structure contains the Db2 shared communications area (SCA). Each member uses the SCA to pass control information to the rest of the members in the group. The SCA contains all database exception status conditions and other information that is necessary for recovery of the group.

**Cache structures (Db2 group buffer pools)**

The cache structures are used as group buffer pools (GBPs), caching shared data pages for the members. You have the following options for caching data of interest to more than one member of a group:

- Cache all data (read-only and updated)
- Cache only data that is updated
- Cache only system control pages (specific to LOB table spaces)
- Cache no data; use the group buffer pool only for cross-invalidation
Group buffer pools use a cross-invalidation mechanism to maintain data consistency across the buffer pools of group members. Shared data pages are registered in a group buffer pool directory in each cache structure, thus enabling the coupling facility control program to cross-invalidate the copies of data pages that are held in individual member buffer pools. Cross-invalidation takes place when a member’s own buffer pool does not contain the latest version of the data. In this case, the member must reread the pages from either the group buffer pool or disk, if that member needs to reference the pages again.

If the operating system is z/OS V2R2 or later, and [z/OS APAR OA54688](http://www.ibm.com) is applied, Db2 can perform cross-invalidation synchronously or asynchronously to the completion of the cache request in the coupling facility. Db2 requests asynchronous cross-invalidation of data pages for table spaces that are defined with page-level locking or above, and when cross-invalidation is done for page updates during commit operations. Although data sharing groups whose members are far from each other can experience more substantial performance improvement with asynchronous cross-invalidation, data sharing groups whose members are close together can also benefit.

One group buffer pool exists for all member buffer pools of the same name. For example, each member must have a buffer pool 0 (BP0) that contains the catalog and directory table spaces. A group buffer pool 0 (GBP0) must exist on a coupling facility for that data sharing group.

Similarly, if a member creates table space X and associates it with buffer pool 1 (BP1), X is associated with BP1 for every member because there is only one definition of X in the catalog for the entire group. To share the data in X, you must define the cache structure, group buffer pool 1 (GBP1). If you do not define the group buffer pool, a single member can update X or more than one member can read X, but there can be no inter-Db2 read/write activity for X.

**Recommendation:** For data that is private to each member, such as work files or user data that only one member reads, define that data to a buffer pool for non-shared page sets. For example, assume that you want to associate all non-shared page sets with buffer pool 6 (BP6). If you want only member DB2A to access a non-shared table space Y, then define Y (and any indexes) to BP6. Define BP6 with a size of 0 (zero) and then you do not need to define the coupling facility structure for group buffer pool 6. By moving private data to buffer pools that are separate from buffer pools used by shared data, you can more easily monitor, and provide for more predictable performance of, private data.

**Defining coupling facility structures:**

Before you enable Db2 data sharing, you must define coupling facility structures.

**About this task**

You must define one lock structure, one list structure, and at least four cache structures:

- Group buffer pool 0
- Group buffer pool 8K0
- Group buffer pool 16K0
- Group buffer pool 32K
The lock structure and list structures (SCAs) do not need to be in the same coupling facility. Individual structures cannot span coupling facilities.

**Recommendation:** Use a Sysplex Failure Management (SFM) policy to define the availability characteristics of the coupling facility structures for lost connectivity failures, which includes a total failure of the coupling facility.

**Procedure**

To define coupling facility structures:

1. Determine the following characteristics of your Db2 coupling facility structures:
   
   **Initial size and maximum size of the structures**
   The structures can be dynamically resized from INITSIZE up to the value in SIZE.

   **Structure names**
   Determine the names that you will use for the coupling facility structures.

   **Availability characteristics**
   You must know the preference list (PREFLIST) for rebuilding or reallocating a structure, if the coupling facility fails.

2. Use the z/OS coupling facility resource management (CFRM) policies to define these structures to the Parallel Sysplex and define a SFM policy.

   A CFRM policy determines how and where the structure resources are allocated.

   See [Adding MVS systems to a sysplex](#) for information about how to create CFRM and SFM policies.

**Example**

A sample CFRM policy is shown in the following figure.
// POLICYX JOB MSGCLASS=Z, REGION=2000K, CLASS=A,
// MSGLEVEL=(1,1)
// STEPLIB EXEC PGM=IXCMIAPU
// STEPLIB DD DSN=SYS1.MIGLIB, DISP=SHR
// SYSPRINT DD SYSOUT=*  
DATA TYPE(CFRM) REPORT(YES)
DEFINE POLICY NAME(POLICYX) REPLACE(YES)
  STRUCTURE NAME(DSNDB0A_LOCK1)
    INIT_SIZE(32000)
    SIZE(64000)
    REBUILD_PERCENT(5)
    PRELIST(CF01, CF02)
  STRUCTURE NAME(DSNDB0A_GBP0)
    INIT_SIZE(50000)
    SIZE(100000)
    REBUILD_PERCENT(5)
    DUPLEX(ALLOWED)
    PRELIST(CF02, CF01)
  STRUCTURE NAME(DSNDB0A_GBP1)
    INIT_SIZE(50000)
    SIZE(100000)
    PRELIST(CF02, CF01)
    DUPLEX(ENABLED)
  STRUCTURE NAME(DSNDB0A_SCA)
    INIT_SIZE(10000)
    SIZE(20000)
    REBUILD_PERCENT(5)
    PRELIST(CF01, CF02)

CF NAME(CF01) TYPE(009674)
  MFG(IBM)
  PLANT(00)
  SEQUENCE(000000040016)
  PARTITION(1)
  CPCID(00)
  DUMPSPACE(1200)

CF NAME(CF02) TYPE(009674)
  MFG(IBM)
  PLANT(00)
  SEQUENCE(000000040029)
  PARTITION(1)
  CPCID(00)
  DUMPSPACE(1200)

//  
Figure 83. Sample CFRM policy

Related concepts:
- Coupling facility availability
- Coupling facility structure names
- Duplexed structures
- Storage estimates for data sharing environments
- How to change structure sizes
- How structures are rebuilt when connectivity is lost

Authorization for Db2 to access coupling facility structures:

Optionally, you can set up a facility class profile to limit access to the structures in the coupling facility.
If you do this, ensure that Db2 does have access by ensuring that the IDs that are associated with the Db2 address spaces have alter access authority to the coupling facility structures through RESOURCE(IXLSTR,structure_name) in SAF class CLASS(FACILITY).

If you do not create a facility class profile, the default allows any authorized user or program (supervisor state and program key mask allowing key 0-7) to issue coupling facility macros for the structure.

**Sysplex timer**
Install at least one Sysplex Timer in the Parallel Sysplex.

For high availability, more than one Sysplex Timer is required. The Sysplex Timer synchronizes the timestamps of the IBM System z9® or System z10® and IBM Z servers for all members of the data sharing group. Db2 data sharing uses a value that is derived from the timestamp (as seen in the log) to recover data.

**Common z/OS libraries**
Db2 supports a configuration with a SYS1.PARMLIB and SYS1.PROCLIB that is shared by all z/OS systems in the Parallel Sysplex. This configuration lets you add and modify systems more easily.

If you intend to have many members in the Parallel Sysplex, each Db2 and IRLM that you define to the z/OS system in the IEFSSNxx parmlib member requires a z/OS system linkage index (LX). The default number of these indexes that z/OS reserves is 165. If you place all of your Db2 and IRLM subsystem definitions in a single IEFSSNxx member, you might need more than 165 LXs to start the members.

If you need more than 165 LXs, use the NSYSLX option on the z/OS IEASYSxx parmlib member to increase this number. See [z/OS MVS Initialization and Tuning Reference](#) for more information.

**Related tasks:**
- Naming in Db2 data sharing

**Connectivity requirements in a Parallel Sysplex**
Db2 data sharing requires that all Db2-related resources reside on shared disks. These resources include the Db2 catalog and directory and any user data, and the integrated catalog for Db2 data sets.

Also, all the members’ logs and bootstrap data sets (BSDSs) must be on shared disks for recovery purposes. A member that performs recovery must have access to the logs of other members in the group.

**Recommendation:** Place work files on shared disks for the following reasons:
- A member stays connected to its work file even if you need to restart the member on another processor.
- You can create or drop a work file table space from any other member in the data sharing group.

Ensure that you have physical connectivity by checking the following connections:
- Verify that one user-integrated catalog facility exists for cataloging the data sets of a data sharing group, and that you can access this catalog from each z/OS system in the Parallel Sysplex.
- Verify connectivity to the following entries from each system on which a member resides:
A set of Db2 target libraries
A single Db2 catalog
A single Db2 directory
All databases that are shared
All log data sets
All BSDS data sets
All coupling facilities used by the data sharing group
User integrated catalog facility catalogs for shared databases

Naming in Db2 data sharing

By maintaining a consistent naming convention, you can simplify the process of naming the many items in a Db2 data sharing environment. You also simplify the management of name changes, as you add new members or move existing members to different systems in the future.

About this task

You assign to names to entities during the process of installing Db2 and enabling data sharing. Some parts of a data sharing system require you to specify a valid name. Other names are generated during the installation process.

After installation, you cannot change some names, such as the group name and member names. To change names of data sharing members you must quiesce existing members and add replacement members with the new name. For more information about that process, see Renaming data sharing members.

Assign names to both IRLM and data sharing groups, and to members within a group. One recommendation is to make names and prefixes unique within the Parallel Sysplex. Although this uniqueness is not required for all names, it helps you avoid problems with identifying and moving entities among z/OS systems in the Parallel Sysplex.

If you are enabling the originating member of the group from an existing Db2 subsystem, you can build a naming scheme around existing names to reduce the disruption to existing applications. However, before enabling data sharing, you might want to change some names to lay the foundation for a solid naming convention. For information about the process for changing the name of the originating member, see Renaming the originating Db2 subsystem for a new data sharing group.

Another name to choose carefully is the catalog alias for the group. It is very difficult to change that name. To change the catalog alias for the group, you must bring the entire group down and perform the single-system procedure for every member of the group.

Procedure

Consider the following suggestions when naming various Db2 entities:
1. Name subsystems first and use the subsystem names as the basis for creating other names. Subsystem names are limited to four characters, making them the shortest names.
   Four-character names are also needed for the following entities:
   • Db2 group name; one per data sharing group
   • Member name; one for each Db2 subsystem in the group
• IRLM subsystem name; one for each member of the group

One possible naming convention is a Parallel Sysplex-wide approach to naming. This convention assigns subsystem names of the form “ctmg” where:

- **c** Denotes a particular collection of logically related applications or subsystems.
- **t** Denotes a type of resource, such as “B” for Db2 or “J” for IRLM.
- **m** Denotes a particular member within a data sharing group or an IRLM group. This identifier also associates a subsystem with its z/OS system for recovery when you use automatic restart.
- **g** Denotes a particular Db2 group.

This information uses “A” as the group identifier. Therefore, the naming scheme has a subsystem name of DB1A for the first member of the data sharing group. The second member is DB2A, and so on. The following table shows how the member identifier relates to a particular z/OS system and how the group identifier associates members across a Parallel Sysplex system. In each column are the names that are associated with a particular z/OS recovery group (z/OS2, for example). In each row are the names that are associated with the primary components of the data sharing group, such as Db2 subsystems and IRLM subsystems.

<table>
<thead>
<tr>
<th>Table 57. Subsystem names in a Parallel Sysplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS1</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>DA11 (CICS)</td>
</tr>
<tr>
<td>DB1A (Db2)</td>
</tr>
<tr>
<td>DJ1A (IRLM)</td>
</tr>
<tr>
<td>DD11 (DBCTL)</td>
</tr>
</tbody>
</table>

2. If you enable an existing Db2 subsystem for data sharing, and existing applications already access that Db2 subsystem, consider using the name of the existing Db2 subsystem as the group attachment name. This strategy allows existing applications to continue to use the data sharing member. It does not require any changes to those applications in terms of specifying a new subsystem name.

3. Use the same name for the Db2 group name, the Db2 location name, and the Db2 integrated catalog alias. A data sharing group uses the same catalog alias name. This catalog alias is used as the high-level qualifier for the Db2 directory (DSNDB01), catalog (DSNDB06), default database (DSNDB04), and work file database VSAM data sets.

4. Use the same names as the member name and Db2 subsystem name for each member.

5. Accept the default command prefix for each member.

**Example of names for a Db2 data sharing environment**

<table>
<thead>
<tr>
<th>Table 58. Db2 data sharing naming example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
</tr>
<tr>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Db2 group name</td>
</tr>
<tr>
<td>ICF catalog alias</td>
</tr>
<tr>
<td>Group attachment name</td>
</tr>
<tr>
<td>Subgroup attachment name</td>
</tr>
</tbody>
</table>

Chapter 9. Installing, migrating, and enabling Db2 data sharing

609
**Table 58. Db2 data sharing naming example (continued)**

<table>
<thead>
<tr>
<th>Entity</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog and directory database names</td>
<td>DSNDB0A.DSNDB01 and DSNDB0A.DSNDB06</td>
</tr>
<tr>
<td>Application defaults module</td>
<td>DSNHDECP</td>
</tr>
<tr>
<td>Target libraries</td>
<td>DSNDB0A.SDSNCLST DSNDB0A.SDSNLINK DSNDB0A.SDSNLOAD DSNDB0A.SDSNEXIT</td>
</tr>
<tr>
<td>Network Address</td>
<td>DOMAIN DSNDB0A.svl.ibm.com AND PORT 446</td>
</tr>
<tr>
<td>Group DRDA Port</td>
<td>446</td>
</tr>
<tr>
<td>Group Secure DRDA Port</td>
<td>448</td>
</tr>
<tr>
<td>Location name</td>
<td>STLEC1</td>
</tr>
<tr>
<td>Generic LU name</td>
<td>USIBMSY.SYEC1GLU</td>
</tr>
<tr>
<td>Parallel Sysplex domain name</td>
<td>DSNDB0A.svl.ibm.com</td>
</tr>
<tr>
<td>Group IPv4 Address</td>
<td>1.1.1.1</td>
</tr>
<tr>
<td>Group IPv6 Address</td>
<td>1.1.1.1.1.1.1.1</td>
</tr>
<tr>
<td>Member name</td>
<td>DB#A</td>
</tr>
<tr>
<td>Db2 subsystem name</td>
<td>DB#A</td>
</tr>
<tr>
<td>Location alias name</td>
<td>ALIAS1</td>
</tr>
<tr>
<td>Member LU name</td>
<td>LUDB2LA</td>
</tr>
<tr>
<td>Member domain name</td>
<td>DSNDB0A.svl.ibm.com</td>
</tr>
<tr>
<td>Network address</td>
<td>DOMAIN DSNDB0A.svl.ibm.com AND PORT 5001</td>
</tr>
<tr>
<td>Resync Port</td>
<td>5001</td>
</tr>
<tr>
<td>Member IPv4 Address</td>
<td>2.2.2.2</td>
</tr>
<tr>
<td>Member IPv6 Address</td>
<td>2.2.2.2.2.2.2.2.2</td>
</tr>
<tr>
<td>Command prefix</td>
<td>-DB#A</td>
</tr>
<tr>
<td>Work file database</td>
<td>WRKDB#A</td>
</tr>
<tr>
<td>Load module for subsystem parameters</td>
<td>DSNZP0#A</td>
</tr>
<tr>
<td>Member BSDS names</td>
<td>DSNDB0A,DB#A.BSDS01 DSNDB0A,DB#A.BSDS02</td>
</tr>
<tr>
<td>Active log data set prefixes</td>
<td>DSNDB0A,DB#A.LOGCOPY1 DSNDB0A,DB#A.LOGCOPY2</td>
</tr>
<tr>
<td>Archive log data set prefixes</td>
<td>DSNDB0A,DB#A.ARC1 DSNDB0A,DB#A.ARC2</td>
</tr>
<tr>
<td>SCA and group buffer pool connection names</td>
<td>DB2_DB#A</td>
</tr>
<tr>
<td>Procedure names</td>
<td>DB#AMSTR DB#ADBM1 DB#ADIST DB#ASPAS</td>
</tr>
<tr>
<td>XCF member name</td>
<td>DB0A</td>
</tr>
<tr>
<td>IRLM group name</td>
<td>DXRDB0A</td>
</tr>
<tr>
<td>IRLM subsystem name</td>
<td>DJ#A</td>
</tr>
</tbody>
</table>
Table 58. Db2 data sharing naming example (continued)

<table>
<thead>
<tr>
<th>Entity</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRLM procedure name</td>
<td>DB#AIRLM</td>
</tr>
<tr>
<td>IRLM member ID</td>
<td>Number 1-12 corresponding to #</td>
</tr>
<tr>
<td>IRLM XCF member name</td>
<td>DXRDB0A$$DJ#A12</td>
</tr>
<tr>
<td>Lock structure connection name</td>
<td>DXRDB0A$$DJ#A12</td>
</tr>
<tr>
<td>Cache structure (group buffer pools)</td>
<td>DSNDB0A_GBP0</td>
</tr>
<tr>
<td>Lock structure</td>
<td>DSNDB0A_LOCK1</td>
</tr>
<tr>
<td>List structure (shared communications area)</td>
<td>DSNDB0A_SCA</td>
</tr>
</tbody>
</table>

What to do next

Consider a convention for names for distributed processing. Those naming conventions are probably be part of a much broader convention. See your network administrator for more information about choosing names for distributed access.

Data sharing group names

Group names are shared by all members of the data sharing group. Db2 group names include the generic LU name, group attachment name, location name, and coupling facility structure names.

The following names are considered group names:

Db2 group name

The name that encompasses the entire data sharing group. The coupling facility structure names are based on this name.

The Db2 group name must be unique within the Parallel Sysplex. If you use this name as a basis for the location name, the Db2 group name must be unique within the network.

This name can be up to eight characters long, it must begin with an alphabetic character, and it can consist of the characters A-Z, 0-9, $, #, and @. An example of a Db2 group name is DSNDB0A.

Restrictions: To avoid names that IBM uses for its XCF groups, do not begin Db2 group names with the letters A-I unless the first three characters are DSN. Do not use the string SYS as the first three characters, and do not use the string UNDESIG as the group name.

Important: Never reuse a Db2 group name, even if a data sharing group that previously used the name no longer exists. Some data sharing information, such as the Db2 group name, is retained in the Parallel Sysplex couple data set (CDS). To determine what Db2 group names exist, execute the z/OS command DISPLAY XCF, GROUP.

ICF catalog alias

The name of the Integrated Catalog Facility (ICF) catalog alias that you must place in the z/OS master catalog. This name can be up to 8 characters long, and it should be the same as the Db2 group name. An example of an ICF catalog alias name is DSNDB0A.

Group attachment 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. This name can be up to four characters long. An example of a group attachment name is DB0A.

**Subgroup attachment name**
The name that is used by TSO, CAF, RRSAF, JDBC, ODBC, and Db2 utilities connections requests to identify which subgroup attachment to use. This name can be up to four characters long. An example of a subgroup attachment name is SBG1.

**Catalog and directory database names**
All Db2 subsystems in the group share the same catalog and directory. Examples of the catalog and directory database names are catalias.DSNDB01 and catalias.DSNDB06.

**Application defaults module**
The name of the module that contains the application programming defaults. This module resides in SDSNEXIT. The name can be up to eight characters long. DSNHDECP is an example module name.

**Target libraries**
Target libraries can be shared among Db2 group members or can be replicated. Example names for target libraries are catalias.SDSNCLST, catalias.SDSNLINK, catalias.SDSNLOAD, and catalias.SDSNEXIT.

**Network Address**
The group DVIPA and port number. It can be an IPv4 address, an IPv6 address, or both.

**DRDA port number**
The port number that is used by the distributed data facility (DDF) to accept incoming SQL connection requests, if the data sharing group uses TCP/IP. 446 is the recommended DRDA port.

**Secure DRDA port number**
The secure port identifies the TCP/IP port number that is used by DDF to accept inbound secure DRDA connection requests.

**Location name**
The name that is used when the data sharing group processes distributed requests. The group is treated as a single location by remote requesters. Requesters use this name in their SQL applications. This name can be up to 16 characters long.

**Generic LU name**
This name lets remote requesters configure their systems to treat the data sharing group as a single LU. This name can be up to eight characters long.

**Parallel Sysplex domain name**
This name lets you take advantage of workload balancing for TCP/IP connections.

**Group IPv4 address**
GRPIPV4 identifies a constant IPv4 address to be associated with the data sharing group for which this DDF is a member.

**Group IPv6 address**
GRPIPV6 identifies a constant IPv6 address to be associated with the data sharing group for which this DDF is a member.

**Related concepts:**
Group attachment names and subgroup attachment names

Group and subgroup attachments are used by TSO, CAF, RRSAF, JDBC, ODBC, and Db2 utilities connections to find active Db2 subsystems.

A group attachment name acts as a generic name for all the members of a data sharing group. Subgroup attachment names can be used to specify a subset of members within a group attachment and provide more organization and control over connection requests. To specify subgroup attachment names, you must use Db2 12 early code.

Specify the group attachment name and subgroup attachment name for a subsystem in member IEFSSNxx. You can let the Db2 installation process do this for you by specifying the names on installation panel DSNTIPK, or you can update the member yourself. If you use the DSNTIPK panel, the installation CLIST places the specified names in the IEFSSNxx member and in the DSNHDECP load module for the group. Specify the group attachment name and subgroup attachment name at a convenient time (during a planned IPL, for example).

The following rules apply to group attachment and subgroup attachment names:

- A subgroup attachment cannot have the same name as a group attachment.
- Neither the group attachment nor the subgroup attachment should have the same name as a Db2 member.
- If you specify a subgroup attachment name, you must also specify a group attachment name. A subgroup attachment must belong to one, and only one group attachment.
- A member can belong to one subgroup attachment at most and does not need to belong to a subgroup attachment.

The group attachment name and subgroup attachment name are included in the output for the command DISPLAY GROUP.

Even if you have not yet enabled data sharing, the group attachment name and subgroup attachment name is active after you IPL the system. An active group attachment name and subgroup attachment name in a non-data sharing environment is not a problem. Until you are ready to move to data sharing, continue to specify the Db2 subsystem name in your TSO and batch jobs. When you are ready to move to data sharing, you can change those jobs to specify a group attachment name and subgroup attachment name without the need for an IPL.

How Db2 chooses a subsystem name

When you submit a job on a z/OS system, Db2 treats the name that you specified on the Db2 connection request as a subsystem name, a group attachment name, or a subgroup attachment name. Db2 first assumes that the name is a subsystem name and attaches to that subsystem if either of the following is true:
• The subsystem is started.
• The subsystem is not started, and NOGROUP was specified in the Db2 connection request. NOGROUP indicates that group attachment or subgroup attachment processing is not to be considered. If RETRY was specified in the command, Db2 tries to attach the subsystem again in 30 seconds. The value of RETRY determines the number of times that Db2 re-attempts to attach.

Db2 assumes that the name on the Db2 connection request is a group attachment name or subgroup attachment name if no qualifying subsystem is found and either of the following is true:
• No subsystem with the name in the command is defined.
• A subsystem with that name is not started, the group attachment name or subgroup attachment name is the same as its subsystem name, and NOGROUP was not specified in the Db2 connection request.

When Db2 assumes that the name is a group attachment name or subgroup attachment name, it performs the following actions:
• Constructs a list of Db2 subsystems that are defined to this z/OS. To create the list, Db2 adds each subsystem during subsystem initialization. At IPL time, subsystems are initialized in the order in which they appear in member IEFSSNxx if the MVS Subsystem Interface (SSI) is running on a single processor. If the MVS SSI is running on multiple processors in parallel, the subsystems are placed in the group attachment name list in the order that they complete initialization. In this case, the order of the list is unpredictable. If you add a subsystem with the z/OS SETSSI command, that subsystem is added to the list at that time.
• Tries to attach to the subsystems in the list in random order until it finds one that is started on this z/OS. However, if RANDOMATT=NO has been specified for a Db2 subsystem, that subsystem is excluded from the random search.
• Searches for any available Db2 subsystem within the group or subgroup, including those that specify RANDOMATT=NO, if the random search finds no available Db2 subsystems or if RANDOMATT=NO is specified for all subsystems. Search order is defined according to the order in which z/OS subsystem initialization completes for each Db2 subsystem. If you want to direct work to certain Db2 subsystems, set the RANDOMATT subsystem parameter to YES on those subsystems. If you want to direct work away from certain Db2 subsystems, set the RANDOMATT subsystem parameter to NO on those subsystems.

If the name on the Db2 connection request is not a group attachment name or subgroup attachment name, a “not started” message is returned.

If you need to prevent a batch workload from running on a Db2 subsystem that is intended for high-performance online transactions, specify RANDOMATT=NO for the online subsystems and RANDOMATT=YES for the batch workload subsystems. In that case, the batch jobs use the group attachment name or subgroup attachment name, and the online subsystem does not.

Alternatively, to achieve non-randomized group attachments or subgroup attachments, as in Version 8 of Db2, specify RANDOMATT=NO for all members of the group.
When a subsystem and a group attachment name or subgroup attachment name are the same

When you begin moving to data sharing, ensure that your IEFSSNxx definitions are correct and Db2 connection requests are coded to get the results that you intend. You should especially ensure that the definitions are correct when the group attachment name or subgroup attachment name is the same as a subsystem name. Incorrect IEFSSNxx definitions might be troublesome if you have inactive subsystems that are still defined but not used.

Assume you have the following subsystem definitions on a z/OS system:

```
DB2A,DSN3INI,'DSN3EPX,-DB2A,S'  +Inactive subsystem
DB1A,DSN3INI,'DSN3EPX,-DB1A,S,LIB2A'  +Active subsystem
```

The jobs that are submitted on this z/OS system try to connect to the name DB2A. Db2 tries to connect to subsystem DB2A before considering group attachment processing. However, because DB2A is not started and it lacks a group attachment name, Db2 does not invoke group attachment processing and find DB1A as you might have intended. To avoid this situation, include the group attachment name in the definition of DB2A, or remove entry IEFSSNxx for subsystem DB2A if it is obsolete. With DB2A defined as the group attachment name for subsystem DB2A, Db2 tries to attach to DB1A after it discovers that DB2A is not started.

Alternatively, you might want a job to connect to a specific subsystem, but it connects to another subsystem in the group instead.

Assume that you have the following subsystem definitions on a z/OS system:

```
DB1A,DSN3INI,'DSN3EPX,-DB1A,S,LIB1A'. +Inactive subsystem
DB2A,DSN3INI,'DSN3EPX,-DB2A,S,LIB2A'. +Inactive subsystem
DB3A,DSN3INI,'DSN3EPX,-DB3A,S,LIB3A'. +Active subsystem
```

Notice that DB1A is specified as a subsystem and also as the group attachment name for all three subsystems.

The jobs are submitted on this system with a Db2 connection request that specifies the name DB1A and omits the NOGROUP keyword. Db2 tries to connect to subsystem DB1A, but does not. Instead, Db2 invokes group attachment processing and eventually connects to DB3A, the first active subsystem with the group attachment name. You might have intended that the job connect only to DB1A. To ensure that Db2 connects to subsystem DB1A and no other subsystem, specify NOGROUP in the Db2 connection request to disable group attachment processing. Specify the RETRY keyword in the request to indicate the number of times, at 30-second intervals, that Db2 will try to connect to DB1A again.

Related tasks:
- Registering the command prefixes, member group attachment name, and subgroup attachment name

Member names

Member names include the individual Db2 member (subsystem) name and its associated z/OS subsystem name, procedure names, and BSDS and log names. Member names must be unique within the data sharing group or, in certain cases, the z/OS Parallel Sysplex.

The following names are considered member names:
Member name
The name of an individual member of a data sharing group. A member name can be up to eight characters long, and it can consist of the characters A-Z, 0-9, $, #, and @. Db2 uses this name to form its z/OS cross-system coupling facility (XCF) member name. This name must be unique within the data sharing group. If the member name is also used as the high level qualifier for the member's data sets (BSDS, logs, and so on), member names must be unique within the z/OS Parallel Sysplex. This is because the z/OS Parallel Sysplex can have a shared master catalog. An example of a member name is DB1A.

Db2 subsystem name
The name of a Db2 subsystem that is a member of a data sharing group. This name is used by all the attachment interfaces. A Db2 subsystem name can be up to four characters long, and it must be unique within the Parallel Sysplex. The member name and the Db2 subsystem name should be the same. An example of a Db2 subsystem name is DB1A.

Location alias name
This name can represent one, several, or all members of a data sharing group. A location alias name enables you to define subsets of data sharing group members. Subsetting gives you the ability to limit the members to which DRDA requesters can connect.

Member LU name
The network name of an individual member of a data sharing group. A logical unit (LU) name must be unique both within the group and within the network. This name can be up to eight characters long.

Member domain name
This name lets Db2 handle indoubt thread resolution for TCP/IP connections.

Network Address
The member DVIPA and port number. It can be either an IPv4 address, an IPv6 address, or both. The TCP/IP port number that is used by the distributed data facility (DDF) to accept incoming DRDA two-phase commit resynchronization requests, if the data sharing group uses TCP/IP. Each member of a data sharing group listens on a unique resynchronization port.

Member IPv4 Address
A constant IPv4 address to be associated with the data sharing member.

Member IPv6 Address
A constant IPv6 address to be associated with the data sharing member.

Command prefix
The prefix that directs commands entered at a z/OS console to a particular member of a data sharing group. The command prefix can be up to eight characters long and the first character must be a special character. The default prefix is the concatenation of the hyphen character (-) with the Db2 subsystem name.

This string is specified as a parameter on the IEFSSNxx subsystem definition. An example of a command prefix is -DB1A. You can have blanks between the command prefix and the command.

Do not assign a command prefix that is used by another Db2 subsystem or that can be interpreted as belonging to more than one Db2 subsystem or z/OS application. Specifically, do not specify a multiple-character
command prefix that is a subset or a superset of another command prefix starting from the first character. For example, it is invalid to assign a hyphen (-) to one Db2 subsystem and '-DB2A' to another Db2 subsystem. Similarly, it is invalid to assign '?Db2' to one Db2 subsystem and '?DB2A' to another Db2 subsystem. It is valid, for example, to assign '-DB2A' and '-DB3A' to different Db2 subsystems.

Work file database
The name of the work file database that is associated with a member of the data sharing group. Each member has its own work file database. In a non-data sharing environment, a Db2 subsystem's work file database is called DSNDB07. In a data sharing environment, although one member of the group can have a work file database with the name DSNDB07, you might want to create one with a more meaningful name, such as WRKDB1A, for member DB1A. This name can be up to eight characters long.

Work file data sets have names of the format catalias.DSNDBC.mworkdb.DSNkknn.y0001.A001, where y can be either I or J.

Restriction: You cannot specify a work file database name that begins with DSNDB unless the name is DSNDB07.

Load module for subsystem parameters
The name of the load module that contains a member's subsystem parameters. Each member has its own subsystem parameters. The load module resides in SDSNEXIT. The name is specified as a parameter on mssnMSTR procedure.

The load module name can be up to eight characters long. An example name is DSNZP01A, a naming convention that you can use to associate member DB1A with DSNZP01A.

Choosing names for member data sets: When choosing names for member data sets, remember that data set names beginning with membname must have a master catalog alias to point to the catalog where the data sets are cataloged. The Db2 installation process does not create this catalog alias. One way to handle this is to begin member data set names with catalias and a member-related qualifier. For example, member data set names could take the form catalias.membname.xxxxx. This format eliminates the need to have a master catalog alias for membname.

Member BSDS names
The names of bootstrap data sets (BSDS) belonging to a member of a data sharing group. The catalog alias can be added as the first qualifier. These names can be up to 33 characters long. Example BSDS names are DSNDB0A.DB1A.BSDS01 and DSNDB0A.DB1A.BSDS02.

Active log data set prefixes
The prefixes of active log data sets belonging to a member of a data sharing group. The catalog alias can be added as the first qualifier. These prefixes can be up to 30 characters long. Example active log data set prefixes are DSNDB0A.DB1A.LOGCOPY1 and DSNDB0A.DB1A.LOGCOPY2.

Archive log data set prefixes
The prefixes of archive log data sets belonging to a member of a data sharing group. The catalog alias can be added as the first qualifier. These prefixes can be up to 35 characters long unless you want the data sets
timestamped. If they are timestamped, these prefixes can be up to only 19 characters long. Use the TIMESTAMP ARCHIVES parameter of installation panel DSNTIPH to specify whether you want the date and time qualifiers to be added to the archive log data set prefix.

Example archive log data set prefixes are DSNDB0A.DB1A.ARC1 and DSNDB0A.DB1A.ARC2.

SCA and group buffer pool connection names
The connection name is generated by Db2. You see it only in certain commands (z/OS D XCF,STRUCTURE and the Db2 connection list display of DISPLAY GROUPBUFFERPOOL). The connection name can be up to 16 characters long. An example connection name is DB2_membername.

Procedure names
The procedure names are generated from the member subsystem name at installation. The names can be up to eight characters long. Example procedure names are mssnMSTR, mssnDBM1, mssnDIST, and mssnSPAS.

XCF member name
This is the same as the member name for Db2; thus, a maximum of eight characters is used.

Related concepts:

Group access (Db2 Data Sharing Planning and Administration)

Related tasks:
Choosing names for the local subsystem
Registering the command prefixes, member group attachment name, and subgroup attachment name

Related reference:
COMMAND PREFIX field

IRLM names
Each member of a data sharing group has its own IRLM. The IRLM group name, subsystem name, and member ID are parameters on the IRLM startup procedure.

This means that every IRLM in the group must have a separate IRLM procedure. The following figure shows the relationship between Db2 group names and IRLM group names.

Db2 group: DSNDB0A
IRLM group: DXRDB0A

![Diagram showing the relationship between Db2 and IRLM group names]

Figure 84. Relationship between Db2 and IRLM group names

You must choose the following IRLM names before installing Db2:

IRLM group name
The name that encompasses the entire IRLM group. This name can be up to eight characters long and can consist of the characters A-Z, 0-9, $, #, and
@. The group name must begin with an alphabetic character. The name must be unique across the Parallel Sysplex.

**Restriction:** To avoid names that IBM uses for its XCF groups, do not begin with the letters A-I unless the first three characters are DXR. Do not use the string SYS as the first three characters, and do not use the string UNDESIG as your group name.

The IRLM group name is a parameter (IRLMGRP=) on each member’s IRLM procedure. An example IRLM group name is DXRDB0A, a naming convention that you can use to associate member DB0A with DXRDB0A.

**IRLM subsystem name**
Each IRLM must have a subsystem name that must be unique within the data sharing group and can be up to four characters long and. This name is a parameter on the IRLM procedure.

A sample subsystem name is DJ1A. The “1A” characters indicate that this IRLM is paired with the member DB1A.

**IRLM procedure name**
Each member of a data sharing group knows its IRLM by the procedure and subsystem names saved in that member’s installation parameter load module. The IRLM procedure name can be up to eight characters long.

**Recommendation:** Use the member subsystem name followed by “IRLM.” An example IRLM procedure name is DB1AIRLM.

**IRLM member ID**
This ID uniquely names an IRLM within a group. It is a parameter on the IRLM procedure and must be unique within the data sharing group. It is a number between 1 and 255 (inclusive).

The following names also apply to IRLMs:

**IRLM XCF member name**
This name can be up to 16 characters long. An example IRLM XCF member name is xxxxxxxx$issnNNN, where xxxxxxxx is the IRLM group name, issn is the IRLM subsystem ID, and NNN is the IRLM member ID. Dollar signs ($) are used as padding. This name is generated at startup time.

**Lock structure connection name**
The connection name is the same as the IRLM XCF member name. It is generated by IRLM. You see it only as the output of certain commands (such as z/OS D XCF,STRUCTURE). In some cases, the connection name can be of the format xxxxxxxx#$issnNNN.

**Related reference:**

START irlpmpc (z/OS IRLM) (Db2 Commands)

**Coupling facility structure names**
Names for coupling facility structures must conform to a strict naming convention that is based on the Db2 group name.

Sample names are shown in the following table:
Table 59. Sample names of coupling facility structures

<table>
<thead>
<tr>
<th>Structure type</th>
<th>Example name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache structure (group buffer pools)</td>
<td>DSNDB0A_GBP0</td>
</tr>
<tr>
<td>Lock structure</td>
<td>DSNDB0A_LOCK1</td>
</tr>
<tr>
<td>List structure (shared communications area)</td>
<td>DSNDB0A_SCA</td>
</tr>
</tbody>
</table>

**Group buffer pool names**

Use a name of the following format on the CFRM policy to define the group buffer pool to the coupling facility. The name can be up to 16 characters long.

`groupname_GBPxxxx`

Where GBPxxxx is the name of the group buffer pool; the following restrictions apply:

- 4 KB group buffer pools are named GBP0, GBP1, ..., GBP49
- 8 KB group buffer pools are named GBP8K0, GBP8K1, ..., GBP8K9
- 16 KB group buffer pools are named GBP16K0, GBP16K1, ..., GBP16K9
- 32 KB group buffer pools are named GBP32K, GBP32K1, ..., GBP32K9

**Requirement:** You must define GBP0, GBP8K0, GBP16K0, and GBP32K. You must also define any other group buffer pools that you need for your particular data sharing setup.

When Db2 duplexes a group buffer pool structure, the same structure name is used for both the primary and secondary structures. A duplexed structure requires a single CFRM policy structure definition, with one structure name.

**Lock structure name**

Use a name of the following format on the CFRM policy to define the lock structure to the coupling facility. The name can be up to 16 characters long.

`groupname_LOCK1`

**Shared communications area**

Use a name of the following format on the CFRM policy to define the SCA to the coupling facility. The name can be up to 16 characters long.

`groupname_SCA`

**Renaming data sharing members**

Strictly speaking, the renaming of existing data sharing members is not supported. However, you can change the names of your data sharing group members by replacing the members that you want to rename.

**Before you begin**

Changing member names in your data sharing group Db2 data sharing members is an activity that requires careful planning. For more information, see [Naming in Db2 data sharing](#).

**Procedure**

To change the names of existing members of a data sharing group:

1. Define the correct profile names for the DSNR class. Make sure that RACF® definitions are in place to handle the new subsystem names. Add the correct names to the following tables:
2. Replicate existing PERMIT commands to allow users and groups to access the new profiles.

3. Quiesce the member that you want to rename, as described in Quiescing a data sharing member (Db2 Data Sharing Planning and Administration).

4. Add a member with the new name, as described in Adding a new data sharing member.

**Related concepts:**
- Planning for Db2 data sharing

**Related tasks:**
- Renaming the originating Db2 subsystem for a new data sharing group

---

**Renaming the originating Db2 subsystem for a new data sharing group**

When you enable Db2 data sharing, you can rename the Db2 subsystem that becomes the originating member and reuse the original subsystem name as the group attach name. The result is that existing jobs and other workloads can continue to connect to the group instead of the subsystem, without modifications to the jobs or applications.

**Before you begin**

Renaming a Db2 subsystem and renaming members in a data sharing group require careful planning. Also, because every installation has a different configuration, this procedure might not be complete for your environment.

**Tip:** Rename the originating subsystem during the process for enabling Db2 data sharing because Db2 must be stopped when you complete this process.

**About this task**

The following procedure describes an approach to changing the name of the Db2 subsystem that becomes the originating member of a new data sharing group. It is a variation of the process for enabling data sharing, as described in Enabling Db2 data sharing.

The examples change the name of member DB2P to DB1A, which conforms to the naming convention used in other examples in the Db2 data sharing information.

**Procedure**

To rename the originating member when enabling data sharing, complete the following steps:

1. Define the correct profile names for the DSNR class. Make sure that RACF® definitions are in place to handle the new subsystem names. Add the correct names to the following tables:
   - The RACF router table. For more information, see The RACF router table (z/OS Security Server RACF System Programmer’s Guide)
2. Replicate existing PERMIT commands to allow users and groups to access the new profiles.

3. Run the installation CLIST, as described in [Running the CLIST for enabling data sharing](#) and update the following values (you cannot change the subsystem name on the installation panels):
   a. On installation panel DSNTIPH, modify the archive prefix names to include the member name; for example: DB2PCAT.DB1A.ARCLG1
   b. Specify the old subsystem name (for example: DB2P) as the group attachment name.

4. Modify the IEFSSNxx member for this subsystem to include the group attachment name.

   **Tip:** To avoid having to modify JCL for all your jobs, use the existing subsystem name as the group attachment name. This step is necessary to ensure that there are no problems when the call attachment and TSO attachment facilities try to use the group attachment name. For example, if the existing IEFSSNxx member looks like this:

   ```
   DB2P,DSN3INI, "DSN3EPX,?
   ```

   Change it to look like this:

   ```
   DB2P,DSN3INI, "DSN3EPX,?,S,${DB2P}
   ```

5. Add new IEFSSNxx definitions for the new name and prefix by taking one of the following actions:
   - Modify the names to the IEFSSNxx member and IPL z/OS. For example:
     ```
     DB1A,DSN3INI, 'DSN3EPX,-DB1A,S,${DB2P}'
     ```
   - Use the z/OS command SETSSI to set the definitions without an IPL. For example:
     ```
     SETSSI ADD, SUBNAME=DB1A,INITRTN=DSN3INI, INITPARM= 'DSN3EPX,-DB1A,S,${DB2P}'
     ```

   If you use this alternative, you must also add the new name and prefix to the IEFSSNxx member before the next IPL.

    You can leave the old IEFSSNxx definitions in place until you complete the process of enabling data sharing. However, it is best to remove them after you finish enabling data sharing. The change does not take effect until the next IPL of z/OS. However, you can use an SETSSI command to delete the obsolete definition until the next IPL, for example:

    ```
    SETSSI DELETE, SUBNAME=DB2P, FORCE
    ```

6. Stop Db2 and ensure that all work is stopped:
   a. Issue a STOP DB2 command with the MODE(QUIESCE) option.
   b. Restart Db2 in ACCESS(MAINT) mode.
   c. Issue a -DISPLAY THREAD(*) TYPE(*) command to identify indoubt threads. Then use the -RECOVER INDOUBT command for any indoubt threads.
   d. Issue a -DISPLAY UTILITY(*) command to identify unterminated utilities. Then use the -DSN1 TERM UTILITY(*) command to terminate all utilities.
   e. Stop Db2 with the MODE(QUIESCE) option.
7. Rename the BSDS and active log data sets with the new prefix, like DB2PCAT.DB1A.BSDS01.
8. Modify and run job DSNTIJIN to alter the active log data sets to specify SHAREOPTIONS(2 3). Before running the job, verify that the AMS ALTER statements there specify the new name for the active log data sets.
9. Update the BSDS with the renamed log data sets. Include the same RBA ranges as the original active log data sets:
   a. Run the utility print log map (DSNJU004) to obtain the start and end RBAs.
   b. Run the utility change log inventory (DSNJU003) to delete the active logs with the old names.
   c. Run the utility change log inventory (DSNJU003) to add the renamed active logs with the correct ranges. You do not need to add the archive log data sets. The archive log data sets that you specified on panel DSNTIPR replace them.
10. If necessary, increase the size of the BSDS.
11. Run the following jobs to define system settings.
   a. Run job DSNTIJUZ to assemble and link-edit the new subsystem parameter (DSNZPxxx) module. First, verify these settings:
      • DSN6GRP.MEMBNAME: The member name; for example: DB1A
      • DSN6ARVP.ARCPFX1 and DSN6ARVP.ARCPFX2; for example: The archive log data set name prefixes; for example: DB2PCAT.DB1A.ARCHLOG1
      • DSN6SPRM.ADMTPROC: The name of the Db2 Administrative Scheduler address space procedure; for example: DB1AADMT
   b. Run job DSNTIJUA to assemble and link-edit the new application programming default (DSNHDECP) module. First, verify the setting of the DSNHDECM.SSID parameter for the group attach name; for example: DB2P
   c. Run job DSNTIJUL to modify the BSDS and establish the subsystem as the first member of the group. Before running the job, verify the following settings:
      • The names of the BSDS; for example: DB2PCAT.DB1A.BSDS01
      • The GROUPMEM parameter; for example: GROUPMEM=DB1A
12. If necessary, increase the size of the SYSLGRNX table space.
13. Rename the startup procedures. For example, change DB2PMSTR to DB1AMSTR. Do not forget to change the BSDS names to the new names in the ssnmMSTR startup procedure.
14. Modify the IRLM startup procedure for the member.
   a. Change the SCOPE parameter setting from LOCAL to GLOBAL.
   b. Set the IRLMGRP parameter to the name that is to be used for the IRLM group. This name must be different from the Db2 group name.
15. Verify that CICS® can connect to the new Db2 subsystem. There are several ways of doing this, depending on which level of CICS you are running. For example, you might use one of the following techniques:
   • Change the CICS DB2CONN to use the new Db2 subsystem name, reassemble, and link-edit.
   • Modify the JCL for CICS to include the new subsystem name on the INITPARM.
16. Verify that IMS can connect to the new Db2 subsystem.
17. Start Db2 by issuing a START DB2 command.

What to do next

Complete the process for enabling data sharing, by continuing with the step described in [Creating the default storage group and defining temporary work files: DSNTIJTM](#). After data sharing is enabled, remove old definitions from the IEFSSNxIx member, as described at the end of step 5.

Planning for availability in a data sharing environment

When planning your data sharing configuration for the highest availability, the physical protection of the coupling facility and the structures within the coupling facility, as well as the highest network connectivity, are the primary concerns. The SCA and lock structure are both necessary for the group to function.

About this task

To plan for the highest availability:

Procedure

1. Duplex the group buffer pool to allow Db2 to switch to the secondary structure if the primary structure fails.
   
   Although the loss of a group buffer pool does not require a group restart, availability for users and important applications requires that data in a group buffer pool be available as quickly as possible after failure. Group buffer pools have several availability options, depending on the type of failure that occurs. Duplexing group buffer pool structures assures minimal impact on performance. Duplexing can also help you to avoid hours of recovery time, which are often required to recover simplexed group buffer pool structures.
   
   If a simplexed group buffer pool structure fails, the group buffer pool can be recovered automatically from data in the Db2 logs. If members lose connectivity to the group buffer pool, the group buffer pool can be rebuilt in another coupling facility to which the members can connect. Recovering data from the Db2 logs and rebuilding structures in another coupling facility can be very time-consuming.

2. Configure a simplexed SCA and lock structure in failure-isolated coupling facilities. (When duplexed, the SCA and lock structure do not need to be in failure-isolated coupling facilities.)
   
   Duplexing the SCA and lock structure offers marginally faster recovery times compared to dynamically rebuilding a simplexed SCA and lock structure. Duplexing also enables you to use internal coupling facilities (ICFs), which provide cost savings over stand-alone coupling facilities without compromising availability. However, before you decide to duplex the SCA and lock structure, you should consider that:
   
   - Duplexing the SCA and lock structure does not add significant availability, because the structures can be rebuilt very quickly from in-memory data.
   - Duplexing the SCA and lock structure can cause a significant impact on performance. In most cases, the overhead that is incurred by duplexing these structures outweighs the availability benefits of duplexing.

3. Define a dynamic virtual IP address (DVIPA) for the Db2 group and a DVIPA for each Db2 member in the group. DVIPA addressing gives you the ability to
assign a specific virtual IP address to a data sharing group and to each member of the group. This address is independent of any specific TCP/IP stack within the Parallel Sysplex. Even if a member is moved to another z/OS system, as in the case of a failure or maintenance, the member remains accessible and retains the same virtual IP address.

- To achieve the highest level of application availability at the database level, route connections using the Db2 Universal JDBC Driver or Db2 Connect Server with connection concentrator and Sysplex Workload Balancing support enabled. To route all connections through a Db2 Connect Server with Sysplex support enabled, you must first configure the Db2 group and each Db2 member to use DVIPA. The Db2 Connect Server then manages the connections to the group using the DVIPAs.

- To support JTA and JTS distributed transactions, the Db2 Universal JDBC Driver and the WebSphere Application Server can be used to coordinate commits against a data sharing group. If using the Db2 Universal JDBC Driver, group and member DVIPA need to be configured in order to recover indoubt threads after a system failure.

### Automatic restart of z/OS

The purpose of automatic restart is to reduce the amount of time that a particular system is down.

When Db2 or IRLM stops abnormally, the surviving z/OS systems analyze the situation to determine whether z/OS failed too, and where Db2 or IRLM should be restarted. If z/OS did not fail, then Db2 or IRLM is restarted on the same z/OS image. If z/OS failed, then Db2 or IRLM is restarted on another z/OS image.

Automatic Restart Manager (ARM) is the component of z/OS that manages automatic restarts.

### Advantage of automatic restart:

When Db2 and IRLM restart automatically, locks that are held by failed members are released quickly.

Applications that are running on other members can then access data for which the failed member is holding incompatible locks. Consider using automatic restart in conjunction with the RETAINED LOCK TIMEOUT option of installation panel DSNTIPI.

You must install Db2 with a command prefix scope of “started” to take advantage of automatic restart.

### Related concepts:

- The scope and uniqueness of Db2 subsystem parameters in data sharing

### Related tasks:

- Registering the command prefixes, member group attachment name, and subgroup attachment name

### Automatic restart policies:

You control how automatic restart works by using z/OS automatic restart policies.

When the automatic restart function is active, the default action (for both sharing and non-sharing Db2 subsystems and IRLMs) is to restart the subsystems when
they fail. If this default action is not what you want, you must create a policy that defines the action that you want to be taken.

Creating an automatic restart policy:

If the default action of restarting Db2 and IRLM is the action that you want, you do not need to create an automatic restart policy.

Before you begin

If you do want to change the default restart behavior of Db2 or IRLM, you need to know the automatic restart element name.

In a data sharing group, the Db2 element name is the Db2 group name concatenated with the member name (such as DSNDB0ADB1A).

For IRLM, the element name is the IRLM group name concatenated with the IRLM subsystem name and three-character member ID (such as DXRDB0ADJ1A001).

You can also specify a pattern-matching character (such as DSNDB0A*) if you want to use a single policy statement for all members in the group.

About this task

Db2 startup can be a little faster when Automatic Restart Manager (ARM) restarts IRLM because this activity is done in parallel. Db2 does not have to start IRLM and wait. However, if you do not want IRLM to automatically restart after a failure, you can modify the IRLM element.

You can also tailor the restart policy to use restart light. Restart light enables Db2 to restart with a minimal storage footprint to quickly release retained locks and then terminate normally. It is not recommended for a restart in place, but it is recommended for a cross-system restart in the event of a failed z/OS system. It is primarily intended to restart Db2 temporarily on another z/OS system that does not have the capacity to sustain a Db2 and IRLM pair.

Procedure

To create an automatic restart policy:

Create an ARM policy with one or both of the following specifications as needed for your system.

- To specify that Db2 or IRLM is not to be restarted after a failure:
  Include RESTART_ATTEMPTS(0) in the policy for that Db2 or IRLM element.
  For IRLM, you can also use the MODIFY irlmproc,ABEND command, as shown below, to stop IRLM and unregister it from ARM when it comes down.
  Unregistering prevents IRLM from automatically restarting after you bring it down.
  MODIFY irlmproc,ABEND,NODUMP
  However, if a member’s AUTO START option of installation panel DSNTIPI has a value of YES, and if z/OS restarts Db2 automatically, Db2 restarts IRLM, too.

- To have Db2 restarted in a light mode (restart light):
  Modify the ARM policy for the Db2 group to specify LIGHT(YES) or LIGHT(NOINDOUBTS) within the RESTART_METHOD(SYSTEM) keyword for the Db2 element name. For example:
Coupling facility availability

For high availability, you should have at least two coupling facilities, and at least one of them should be non-volatile.

With multiple coupling facilities, you can specify that structures be allocated in the secondary coupling facility, if the primary coupling facility is damaged. You can also consider duplexing SCA, lock, and group buffer pool structures. With duplexing, a secondary structure is always on standby in another coupling facility. This secondary structure is ready to take over if the primary structure fails or if a connectivity failure occurs. If you have three or more coupling facilities, you can even maintain duplexing while performing maintenance on one of the coupling facilities.

Duplexing the group buffer pool structures is a good idea because the performance overhead of duplexed group buffer pools are negligible in most cases, and the availability benefits are very high. Duplexing the SCA and lock structure is not as important for high availability because these structures can be rebuilt dynamically on an alternate coupling facility, if the coupling facility that contains the SCA and lock structure fails.

If the SCA and lock structure are not duplexed, the coupling facility that contains these structures should be failure isolated for the highest availability. A failure-isolated CF resides in a central processor complex (CPC) that does not also contain a data sharing member that is connected to structures in that coupling facility. If the SCA and lock structure reside in a non-failure-isolated coupling facility (a coupling facility that contains the SCA and lock structure and resides in a CPC that also contains a member of that data sharing group), the CPC becomes a single point of failure. If the CPC fails, the entire data sharing group comes down. Duplexing the SCA and lock structure, or keeping the SCA and lock structure in a failure-isolated coupling facility, avoids this single point of failure.

Related concepts:
- Duplexed structures

Related reference:
- [Achieving the Highest Levels of Parallel Sysplex Availability (IBM Redbooks)]
- [z/OS Parallel Sysplex Configuration Overview (IBM Redbooks)]

Recommendations for placement of coupling facilities:

Physically separate the coupling facilities from each other and from the rest of the z/OS images that use those coupling facilities.

If you separate the SCA and lock structures from the systems that use them, you can minimize the chances of performing a lengthy group restart after a lengthy outage.

Quick recovery of the group buffer pools uses information in the lock structure and SCA to determine which databases must be recovered. This is known as damage assessment. Consider putting the lock structure and SCA in a coupling facility that does not contain important cache structures (such as group buffer pool 0). You are less likely to lose the SCA, lock structure, and the group buffer pool at the same time if you carefully separate these structures by placing them in different coupling facilities.
If you lose the lock structure or SCA at the same time as one or more group buffer pools, Db2 waits until the lock structure and SCA are rebuilt before doing damage assessment.

**Preparations for coupling facility failures:**

If structure failures occur, Db2 can recover resources in the structures.

The major types of coupling facility failures are structure failures, connectivity failures, and channel failures:

- A structure failure occurs when a group buffer pool, lock, or SCA structure no longer operates.
- A connectivity failure is a total failure of the coupling facility, such as a power failure to the coupling facility or some problem with the coupling facility control code.
- A channel failure occurs when a channel connecting a CPC to a coupling facility no longer operates.

**Recommendation:** To ensure high availability when preparing for structure and connectivity failures, configure the group buffer pool structures in duplex mode and the SCA and lock structures in simplex mode. To prepare for channel failures, consider using dual channels between each CPC and a coupling facility.

**Duplexed group buffer pool structure failures:**

If a duplexed group buffer pool structure fails, the failing structure is deallocated, and processing continues with the running structure.

This recovery is usually fast and unnoticeable. In case both structures fail, use checkpointing.

**Related concepts:**

- Simplexed group buffer pool structure failures
- How Db2 gathers checkpoint information (Db2 Data Sharing Planning and Administration)

**Simplexed group buffer pool structure failures:**

Simplexed group buffer pools can be recovered from the log when they fail.

Recovery from the log can occur manually, as the result of a START DATABASE command, or it can occur automatically because the group buffer pool is defined with the AUTOREC(YES) option. In either case, to reduce the time that is needed for group buffer pool recovery, use the ALTER GROUPBUFFERPOOL command to make group buffer pool checkpoints more frequent. However, weigh the benefit of faster recovery with the processing resources that are used for the checkpointing. (You can reduce checkpoint costs considerably when z/OS is at the appropriate level of maintenance.

Ensure that you specify one or more alternate coupling facilities in the CFRM preference list for the group buffer pools because a group buffer pool can be allocated in an alternative coupling facility when a new connection is made to it.

**Related concepts:**

- How Db2 gathers checkpoint information (Db2 Data Sharing Planning and Administration)
Problem: Loss of group buffer pool structure (non-duplexed) (Db2 Data Sharing Planning and Administration)

Automatic recovery requirements:

Automatic recovery is faster than manual recovery because Db2 can optimize internal processing of the recovery.

For automatic recovery to be initiated for a group buffer pool, all of the following conditions must be true:

- AUTOREC (YES) is specified for the group buffer pool.
- At least one member is actively connected at the time that the group buffer pool failed. This member must have successfully completed damage assessment.

Db2 never initiates automatic recovery during restart.

For a duplexed group buffer pool, Db2 can use automatic recovery if both instances of the group buffer pool are damaged. Automatic recovery is not needed for group buffer pools that are defined as GBPCACHE (NO).

You can enable automatic recovery with the AUTOREC option of the ALTER GROUPBUFFERPOOL command.

Simplexed SCA and lock structure failures:

To recover a simplexed SCA and lock structure, Db2 uses information that is contained in its virtual storage to quickly rebuild the structures.

This recovery has a minimal impact on performance. Db2 can rebuild a simplexed SCA and lock structure in the same coupling facility or in an alternate coupling facility, assuming that the following conditions are true:

- You specified the alternate coupling facility in the CFRM policy preference list.
- You allocated enough storage in the alternate coupling facility to rebuild the structures there.

If Db2 fails to rebuild the SCA and lock structure from virtual storage, all active members in the group terminate abnormally, and you must perform a group restart to recover the necessary information from the logs.

Duplexed SCA and lock structure failures:

If a duplexed SCA or lock structure fails, the failing structure is deallocated, and processing continues on the running structure.

Important: To enable Db2 to switch to the secondary SCA and lock structure, the CFRM policy for the structures must indicate that duplexing is allowed, and the SCA and lock structure must currently be running in duplexed mode.

Related reference:

- Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)

Preparations for connectivity failures:

Db2 and z/OS interpret a total failure of the coupling facility as a connectivity failure.
Connectivity failures include such issues as power failures to the coupling facility or a problem with coupling facility control code.

Related concepts:
- [Problem: loss of coupling facility (CF) (Db2 Data Sharing Planning and Administration)]

Preparations for coupling facility channel failure:

To prevent a failure to Db2 that is caused by a coupling facility channel failure, consider using dual channels between each CPC and a coupling facility.

Without dual channels (sometimes called links), a channel failure is more likely to occur than a failure in the coupling facility. Losing connectivity to the SCA or lock structure can bring that particular member down, unless you specify duplexing or an alternative coupling facility in the CFRM policy preference list.

Related reference:
- [Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)]

How structures are rebuilt when connectivity is lost:

As with structure failures, Db2 can recover quickly from connectivity failures if group buffer pool structures are duplexed, and if the SCA and lock structure are simplexed.

Recovering from connectivity failures with duplexed structures

Db2 recovers from connectivity failures with a duplexed group buffer pool, SCA, and lock structure by switching to the structure with good connectivity.

Recovering from connectivity failures with simplexed structures

Db2 recovers from connectivity failures with a simplexed group buffer pool, SCA, and lock structure as described below. Be aware that recovery of a simplexed SCA and lock structure is fast and has little or no impact on performance, but recovery of simplexed group buffer pool structures is very disruptive to the system.

When connectivity is lost, Db2 rebuilds simplexed structures on the alternate coupling facility that is specified in the CFRM policy. In rebuilding these structures, Db2 attempts to allocate storage on the alternate coupling facility. Db2 uses the current size of the structure for the initial size of the structure on the alternate coupling facility. If Db2 cannot allocate the storage for the SCA or lock structure, the rebuild fails. If z/OS cannot allocate the storage for the group buffer pools, the changed pages are written to disk.

Related concepts:
- [Failure scenarios (Db2 Data Sharing Planning and Administration)]
- [Problem: a subset of members have lost connectivity (non-duplexed) (Db2 Data Sharing Planning and Administration)]

Specifying when structure rebuilds occur after connectivity is lost:

The rebuild threshold is a percentage that you specify for the REBUILDPERCENT parameter in the CFRM policy. z/OS uses the REBUILDPERCENT value to
determine whether to initiate a structure rebuild in the case of a loss of connectivity to the coupling facility that contains the structure.

**About this task**

The percentage is based on the Sysplex failure management (SFM) weights of all the systems that have active connections to a structure at the time. (A connection is considered active even if connectivity is lost.) You also specify weights on the SFM policy.

z/OS calculates the total weight of:
- All systems that have active connections but that lost connectivity to the structure (A)
- All systems that have active connections to the structure (B)

z/OS then divides A by B and compares the result to the REBUILDPERCENT value. (In the case of multiple connections to a structure from a single z/OS system, that system is counted only once. Multiple connections from a single z/OS system exist, for example, when two members reside on the same z/OS system.)

**Procedure**

To control when a rebuild occurs:

Specify REBUILDPERCENT values in your CFRM policy.

**Recommendation:** If you have high availability requirements, specify a small REBUILDPERCENT value, and allow the system to rebuild the structure. Specify a larger REBUILDPERCENT value only if you prefer to have a member lose the use of a structure rather than temporarily disrupt all of the members that were using the structure so that the structure can be rebuilt.

**Example**

**Example:** All z/OS systems in an eight-member Parallel Sysplex are of equal weight (10), and six of the systems have active connections to the group buffer pool. A equals the total weight of all systems that lose connectivity to the group buffer pool. B equals the total weight of all systems that have active connections to the group buffer pool. If two of the six systems lose connectivity, the value of A is 20, and the value of B is 60.

z/OS determines whether to initiate a rebuild as follows:
- If \((A/B) \times 100\) is greater than or equal to the REBUILDPERCENT value, Db2 rebuilds the structure.
- Otherwise, Db2 does not rebuild the structure. Instead, the affected members disconnect from the group buffer pool.

In this example, \((20/60) \times 100\) is equal to 33. If the value of REBUILDPERCENT is 35, z/OS does not initiate a rebuild.

**Related reference:**

[Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)](chapter9/631)
Rebuild events:

You can monitor a performance class 20 trace (IFCIDs 0267 and 0268) to determine how long a rebuild of a structure takes and the reason for the rebuild.

You can also examine the messages that are returned to the console to monitor a rebuild. The reasons for a rebuild can include:

- Lost connectivity
- Operator command
- Duplexing is being established

Coupling facility volatility:

When a coupling facility is in a volatile state, data in the coupling facility is not saved in the event of a power failure.

Db2 issues a warning message if allocation occurs in a volatile coupling facility. A change in volatility after allocation does not affect existing structures.

If the coupling facility is configured to be non-volatile (using the proper power backup, such as a battery backup), volatility is generally a transient state that might occur, for example, if you take the battery out. If you lose power to a non-volatile coupling facility, the coupling facility enters power save mode and saves the data that is contained in the structures. When power is returned, you do not need to do a group restart nor recover the data from the structures. For systems that require high availability, non-volatile coupling facilities are recommended.

Related concepts:

- When the coupling facility structure fails (z/OS MVS Setting Up a Sysplex)

Duplexed structures

Running some or all of the SCA, lock, and group buffer pool structures in duplex mode is one way to achieve high availability for these structures across many types of failures, including lost connections and damaged structures.

**Important:** Choosing the default duplexing option of synchronous for the lock structure or SCA can have a significant impact on performance. You can improve the performance for duplex mode by choosing the asynchronous duplexing option for the lock structure.

**Important:** If you decide to duplex the SCA and lock structure, you must duplex both structures to achieve the availability benefits. Duplexing only the SCA or only the lock structure does not provide any benefit.

How group buffer pool duplexing works:

With a duplexed group buffer pool structure, two allocations of the same structure use one connection from each member.

Each structure allocation must be in a different coupling facility. z/OS prefers to place the structures in coupling facilities that are failure-isolated from one another.
**Recommendation:** Ensure that at least one of the group buffer pool structures is in a non-volatile coupling facility. If power is lost to both coupling facilities and both coupling facilities are volatile, you must recover the group buffer pool from the logs.

**Characteristics of primary and secondary structures:**

z/OS commands let you stop and start duplexing, and let you choose which of the duplexed structures is the primary and which is the secondary.

The *primary structure* is the one from which changed data is cast out to disk. Db2 also uses the primary structure to read data, keep track of page-registration, and perform cross-invalidation of pages in the buffer pools of individual members. When changed data is written to the primary structure, it is also written to the *secondary structure*. Db2 uses the secondary structure as a backup. After data is cast out to disk from the primary structure, the data is deleted from the secondary structure.

From a z/OS perspective, duplexing is really an extended rebuild, so z/OS documentation and commands sometimes call the primary structure the *old* structure, and refer to the secondary structure as the *new* structure.

**Coupling facility storage considerations for duplexing:**

When planning for storage, make the primary and secondary group buffer pool structures the same size.

If your system is properly configured for high availability using a simplexed group buffer pool, you usually do not need extra coupling facility storage for duplexing. For simplexed structures, you must reserve enough spare capacity in the coupling facilities to be able to absorb the structures of any failed coupling facility. With duplexed structures, instead of reserving storage in case of a coupling facility failure, that storage is used by the secondary group buffer pool.

**Example:** Assume that you have 2 coupling facilities, each with 1 GB of memory, for a total of 2 GB of memory. Ensure that the total size of the structures across the 2 coupling facilities does not exceed 1 GB (50% of the total coupling facility storage).

If you configure three or more coupling facilities, you might need an additional coupling facility for duplexing.

**Duplexing requirements:**

To achieve high availability for SCA, lock, and group buffer pool structures across many types of failures, three conditions must be met.

For duplexing to work, the following conditions must be true:

- The CFRM policy preference list for the group buffer pool structure that you want to duplex must contain at least two coupling facilities with a CFLEVEL of 5 or higher. All members of the data sharing group must have physical connectivity to both coupling facilities in which the primary and secondary structures reside.

If you are going to do automatic reduplexing, you need three coupling facilities that are physically connected to members of the data sharing group.
• At least one member must be actively connected to the group buffer pool structure.
• The group buffer pool must be defined with GBPCACHE(YES), the default.

For asynchronous duplexing of the lock structure, the CFRM couple data set needs to support asynchronous duplexing. To provide that support, add the following item to the CFRM couple data set parameters in the job that runs the format utility:

ITEM NAME(ASYNC DUPLEX) NUMBER(1)

**Important:** Include this item only when the coupling facility level is CFLEVEL 21 or higher, and the z/OS system level is Version 2 Release 2 with the PTF for APAR OA47796, or higher.

**Related reference:**

[Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)]

**Options for duplexing:**

Options for the CFRM policy allow, enable, or disable duplexing.

The following options are available in the CFRM policy for duplexing of the group buffer pool, SCA, and lock.

**Table 60. Duplexing options in the CFRM policy**

<table>
<thead>
<tr>
<th>Option</th>
<th>Valid for</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPLEX(ENABLED)</td>
<td>SCA, group buffer pool, lock</td>
<td>Automatically starts duplexing. For the CF structure, duplexing is started in synchronous mode.</td>
</tr>
<tr>
<td>DUPLEX(ENABLED, duplex-mode)</td>
<td>Lock only</td>
<td>Automatically starts duplexing, with the specified duplex mode. Possible duplex modes are defined below.</td>
</tr>
<tr>
<td>DUPLEX(ALLOWED)</td>
<td>SCA, group buffer pool, lock</td>
<td>Allows duplexing. You must use a command to start duplexing.</td>
</tr>
<tr>
<td>DUPLEX(ALLOWED, duplex-mode)</td>
<td>Lock only</td>
<td>Allows duplexing, with the specified duplex mode. You must use a command to start duplexing. Possible duplex modes are defined below.</td>
</tr>
<tr>
<td>DUPLEX(DISABLED)</td>
<td>SCA, group buffer pool, lock</td>
<td>Disables duplexing</td>
</tr>
</tbody>
</table>

Possible values of *duplex-mode* are:

**SYNCONLY**

The CF structure uses only synchronous duplexing. Either user-managed duplexing or system-managed synchronous duplexing is used. This is the default.
ASYNCONLY

The CF structure uses only asynchronous duplexing. System-managed asynchronous duplexing is used.

When this keyword is used, system-managed duplexing rebuild allocation for the structure requires that the coupling facility level is CFLEVEL 21 or higher.

ASYNC

The structure uses asynchronous duplexing, if possible. If asynchronous duplexing is not possible, the structure uses synchronous duplexing. Some reasons that asynchronous duplexing is not possible are:

• The coupling facility level is less than CFLEVEL 21.
• There is no connector or application support for asynchronous duplexing.

Related tasks:

- Starting duplexing for a structure (Db2 Data Sharing Planning and Administration)
- Stopping duplexing for a structure (Db2 Data Sharing Planning and Administration)

Related reference:

- Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)

Performance implications of duplexing:

The process of establishing duplexing can be somewhat disruptive because access to the group buffer pool structure is quiesced while the secondary structure is allocated. Also, changed pages are copied from the primary structure to the secondary structure (or cast out to disk).

Transactions that need access to the group buffer pool structure during this process are suspended until the process is complete. Because of this disruption, you need to establish duplexing at a time of low activity on the system. How long the process takes depends on how many pages are copied to the secondary group buffer pool structure.

In general, more processor and elapsed time is needed to do duplexed group buffer pool structure writes and castout processing than to do simplex group buffer pool structure writes and castout processing. Workloads that are more update-intensive will probably experience a slight increase in host CPU usage when duplexing is activated. In most cases, the majority of the CPU increase occurs in the Db2 address space. Duplexing can cause a slight increase in the transaction elapsed time. Read performance is unaffected by duplexing.

The statistics and accounting trace classes contain information about structure duplexing.

Estimating the effect of duplexing on CPU usage:

When you enable duplexing, you will see an increase in the CPU usage in the coupling facility that contains the secondary structure. Duplexing should have little or no impact on the CPU usage in the coupling facility that contains the primary structure.
Procedure

To estimate approximately how much the coupling facility CPU usage will increase when you establish duplexing:
1. Determine the amount of coupling facility CPU usage that the simplexed primary structure consumes.
2. Divide the result in half to determine how much coupling facility CPU usage the duplexed secondary structure will consume.

Duplexing rebuild status:

When a group buffer pool structure is duplexed, it is considered to be in an extended rebuild status called a duplexing rebuild.

This activity is reported in IFCIDs 0267 and 0268, along with other reasons for rebuilding.

Db2 for z/OS support for asynchronous CF duplexing

When the coupling facility (CF) is set up in duplex mode, Db2 for z/OS data sharing performance can improve if asynchronous CF duplexing is in effect.

With asynchronous CF duplexing, multi-site data sharing groups that implement duplexing all Db2 CF structures can have continuous availability without significant performance impact after site failures.

Asynchronous CF duplexing for the lock structure has the following prerequisites:
- z/OS Version 2 Release 2, with the PTF for APAR OA47796
- Firmware support for Coupling Facility Control Code (CFCC) with service level 02.16
- CF-to-CF connectivity through coupling links

If those prerequisites are met, you can enable asynchronous CF duplexing by setting the DUPLEX option in the coupling facility resource management (CFRM) policy to one of the following values:
- DUPLEX(ENABLED,ASYNCONLY)
- DUPLEX(ENABLED,ASYNC)
- DUPLEX(ALLOWED,ASYNCONLY)
- DUPLEX(ALLOWED,ASYNC)

See Options for duplexing for details.

Asynchronous duplexing and coexistence mode

When a data sharing group is in coexistence mode with Db2 11 members, asynchronous duplexing of the lock structure cannot be established. When a Db2 11 member joins a data sharing group, the connection to the lock structure is switched to either simplex mode or to synchronous duplexing mode, depending on the DUPLEX option in the CFRM policy for the lock structure. When asynchronous duplexing is no longer active, z/OS issues message IXC522I. The structure can switch back to asynchronous duplexing when there are no active Db2 11 members left in the group.
Switching from asynchronous CF duplexing

You can switch from asynchronous CF duplexing to a simplex lock structure or to synchronous duplexing of the lock structure at any time by changing the CFRM policy for the lock structure, and rebuilding the lock structure to activate the new policy.

Db2 resource availability considerations

Several options for the catalog and directory, for data in group buffer pools, and for Db2 restart can be used for data that requires high-availability.

Db2 resource availability considerations for a data sharing group are basically the same as for a single Db2 subsystem.

Critical Db2 data

Critical Db2 data should be placed behind high-availability storage controllers, such as the IBM Enterprise Storage Server®. Consider placing the catalog and directory behind a 3990 control unit with dual-write capability for hardware duplexing. Another possibility is to use storage controllers with high-availability characteristics (such as controllers that use RAID technology) for critical Db2 data.

Group buffer pool data

Assign data that require high availability to group buffer pools that reside on non-volatile coupling facilities. Use group buffer pool duplexing, which has minimal impact on performance and can avoid hours of recovery time.

Data availability at restart

For faster restarts, take more frequent checkpoints. Checkpoint frequency is the most important factor in Db2 restart time.

If your installation sometimes has problems with units of recovery (URs) that take a long time to back out after a failure, make plans to reroute work to other members of the group. Another solution is to postpone backout processing for those long-running URs until Db2 is up and receiving new work.

Related concepts:

- Postponed backout in a data sharing environment (Db2 Data Sharing Planning and Administration)

XA support for a Sysplex by the IBM Data Server Driver for JDBC and SQLJ

WebSphere Application Server and the IBM Data Server Driver for JDBC and SQLJ can be used to coordinate distributed transactions using XA protocols supported by DRDA.

This type of XA support is only available for transaction managers that use a single-transport processing model. In a single-transport model, a transaction, over a single transport (physical connection), is tied to a member from xa_start to xa_end. The transaction end is followed immediately by xa_prepare(readonly), xa_prepare plus xa_commit or xa_rollback, or xa_rollback. All of this must occur within a single application process. An example of a transaction manager that uses this model is IBM WebSphere Application Server.
Support for the single-transport processing model also includes indoubt transaction recovery where member information for each recoverable transaction is retrieved through xa_recover, which allows xa_commit or xa_rollback to be directed at the specified member.

Important: When a global transaction runs, locks are shared among branches of the global transaction only if those branches run on the same data sharing member. When Sysplex workload balancing is enabled, there is no guarantee that all branches run on the same data sharing member. As a result, an application that uses global transactions and Sysplex workload balancing might experience deadlocks or timeouts.

To use XA support with client applications, you need to set up dynamic virtual IP addresses. A dynamic IP address needs to be configured for the Db2 group, and one address must be set up for each Db2 member in the group. For XA recovery to be processed on any member after failure, information on indoubt XIDs must be obtained from the SCA. The SCA contains the list of indoubt XIDs for the entire group, as well as each member’s DVIPA and the resync port of the member that owns the indoubt thread. XA recovery returns the list of all prepared or heuristically committed threads for the entire group back to the XA Transaction Manager (TM). The IBM Data Server Driver for JDBC and SQLJ uses the information to send the commit decision to the owning member, when the XA TM recovers, and then commits or rolls back indoubt XIDs.

For the recovery process to function properly, you must set up the group DVIPA. The following figure illustrates how the XA recovery process, that is described above, works. After a system failure, XA recover requires the list of indoubt transactions identified by XA XIDs, to be returned to the group. In this example, if member DB1G fails, DB1G will leave an indoubt thread. The state of the thread, as well as related XIDs, is contained in the log of the DB1G member. The DB1G indoubt thread XID is also contained in the group’s XA indoubt list, in the SCA. When WebSphere tries to recover from the system failure, the IBM Data Server Driver for JDBC and SQLJ sends the XA Recover command to any member of the group. The member then returns the group’s list of indoubt XIDs from reading the XA indoubt list in the SCA. When WebSphere commits or rollbacks the indoubt thread, the IBM Data Server Driver for JDBC and SQLJ sends the decision to member DB1G, which resolves the indoubt thread.
Storage estimates for data sharing environments

As part of planning for a data sharing environment, you should estimate the storage requirements for coupling facility structures and for Db2 resources.

Precise estimates for the sizes of coupling facility structures are hard to provide. Every environment is different, and storage allocation is affected by the processor model and the level of coupling facility control code. Use the information in this section to estimate initial size (INITSIZE) values. Depending on how much your workload varies, consider specifying a larger value for SIZE.

For duplexed structures, the SIZE and INITSIZE parameter values apply to both instances of the structure.
The information in this section assumes that all page sets in a particular group buffer pool are defined with the same GBPCACHE attribute. You can put page sets with different GBPCACHE attributes in the same group buffer pool, but you must adjust the formulas accordingly.

**Coupling facility structure size allocation**

When a new coupling facility structure is allocated for a data sharing group, its size is usually taken from the value of the INITSIZE parameter in the CFRM policy.

After the structure is allocated, you can dynamically change its size with the z/OS command SETXCF START,ALTER,SIZE=newsize,STRNAME=strname. The new SIZE value cannot be greater than the SIZE value in the CFRM policy, but it can be smaller than the INITSIZE value. If the coupling facility has enough space, z/OS increases (or decreases) the size of the structure to the new size; the INITSIZE value of the policy remains unchanged.

In most cases, Db2 uses the new SIZE value for any subsequent allocations of the structure instead of the INITSIZE value in the CFRM policy. Any of the following subsequent allocations can use the new SIZE value:

- A group buffer pool or SCA is deallocated and then reallocated
- A secondary structure is allocated for a duplexed group buffer pool, SCA, or lock structure, if duplexing is started after the size of the primary structure was dynamically changed
- Any structure is rebuilt with the z/OS command SETXCF START,REBUILD,STRNAME=strname

The new size is recorded across a restart of Db2 and is used for all subsequent allocations until one of the following events occurs:

- A CFRM policy is started, and the policy has a different INITSIZE value than the size of the structure that was dynamically changed with the SETXCF START,ALTER command
- Another SETXCF START,ALTER command is issued to dynamically change the size of the structure

**Exception:** If a lock structure is deallocated and all the members are down, the INITSIZE value is used. This is a consideration for disaster recovery or for situations where data sharing groups are cloned. During normal operation of a data sharing group, you are unlikely to encounter this situation.

**Related concepts:**

- [How to change structure sizes](#)

**Coupling facility structure sizer:**

The coupling facility structure sizing tool (CFSizer) simplifies the task of estimating coupling facility structure storage size by asking you questions about your existing or planned configuration.

If you are not sure how to answer the questions, the help functions direct you to the source of the data on your system.

After you decide what your structure sizes are, include those values in the CFRM policy definition. See [Adding MVS systems to a sysplex](#) for more information about creating CFRM policies.
Related tasks:

[1] Determine the size of each coupling facility structure (z/OS MVS Setting Up a Sysplex)

Recommendations for coupling facility sizes:

Coupling facility structures contain some static control structures. When a structure is initially allocated, these static structures are allocated to accommodate the potential size of the coupling facility structure.

In other words, the size of the static structures is proportional to the maximum size (the value of the SIZE parameter) of the coupling facility structure. If the SIZE value is much larger than the INITSIZE value, a large percentage of the initial structure's size might be used for these static structures, leaving you with little usable storage space in the structure.

Recommendation: In general, specify a SIZE value that is larger than the INITSIZE value, but limit the SIZE value in the following ways:

- Limit the SIZE value to no more than two to three times the INITSIZE value for the SCA and lock structure.
- Limit the SIZE value to 1.2 - 1.5 times the INITSIZE value for group buffer pools. The SIZE value should not be more than two times the INITSIZE value.

For example, if the INITSIZE value for a group buffer pool is 100 MB, specify a SIZE value of 200 MB or less.

Group buffer pool sizes

The two parts of a group buffer pool are data pages and directory entries. You can specify the ratio of data pages to directory entities in a group buffer pool, or you can use the Db2 default ratio.

Data pages

Data pages (sometimes called data entries) reside in the group buffer pool. The size of a data page is the same as the page size supported by the corresponding Db2 buffer pools (4 KB, 8 KB, 16 KB, or 32 KB).

If you are caching changed data only, you need enough space to cache changed data plus extra space for pages that are frequently referenced. By caching those frequently referenced pages in the group buffer pool, you can decrease the amount of time it takes for any member to refresh that page in its member buffer pool because you avoid the disk I/O.

If you choose GBPCACHE NONE or GBPCACHE SYSTEM, no user data pages are actually stored in the group buffer pool. However, with GBPCACHE SYSTEM, space map pages for LOBs are cached in the coupling facility.

Directory entries

A directory entry specifies the location and status of a page image somewhere in the data sharing group, whether the image is in the group buffer pool or in one of the member buffer pools. Only one directory entry exists for any given page, no matter how many places that page is cached.

The size of a directory entry is approximately 200 bytes, but it varies somewhat based on the size of the data pages and the CFLEVEL you are using.
Specifying a ratio

The amount of space that is allocated for a group buffer pool is determined by the ratio of directory entries to data pages. When you originally define a structure in the CFRM policy for a group buffer pool, you specify its total size. For Db2, the ratio defaults to five directory entries per data page. Later, you can change the ratio with the ALTER GROUPBUFFERPOOL command. The new value that you define takes effect when the group buffer pool is rebuilt or reallocated.

For group buffer pools that are defined with GBPCACHE(NO), ratios are ignored because no data is actually stored in the group buffer pool.

When possible, both a formula and a general guideline are provided to help you estimate the initial sizes and ratios of your group buffer pools. (The exception is for GBPCACHE ALL group buffer pools, for which only a general guideline is provided.)

The formula is not too complex and is likely to be more accurate, assuming that you are fairly confident of the values for the variables in the formulas. Otherwise, use the general guidelines and then adjust your values from there.

Related concepts:
- Determining the correct size and ratio of group buffer pools (Db2 Data Sharing Planning and Administration)

Related reference:
- zSeries Processor Resource/Systems Manager Planning Guide

Storage estimate for group buffer pools that cache changed data:

The size of a group buffer pool is related to the amount of sharing and the amount of updating. You can use a formula to estimate the storage size of group buffer pools that cache changed data.

An estimate must be based on the total amount of member buffer pool storage multiplied by a percentage based on the amount of update activity. As data sharing and updating increases, more pages must be cached in the group buffer pool, and more directory entries are needed to track inter-Db2 interest.

Formula

The following is a formula for estimating storage for group buffer pools that cache changed data:

\[
\begin{align*}
\text{Data\_entries} & = U \times D \times R \\
\text{Data\_MB} & = \text{Data\_entries} \times P / 1024 \\
\text{Dir\_entries} & = \text{Data\_entries} + (U \times VP) \\
\text{Dir\_MB} & = 1.1 \times \text{Dir\_entries} \times 0.2 / 1024 \\
\text{GBP\_MB} & = \text{Data\_MB} + \text{Dir\_MB} \\
\text{RATIO} & = \text{Dir\_entries} / \text{Data\_entries}
\end{align*}
\]

Where:

- \( U \) The estimated degree of data sharing:
  - 1 A high amount of sharing with a lot of update activity
  - 0.7 A moderate amount of sharing with a moderate amount of update activity
A low amount of sharing with a low amount of update activity.

**D**
The number of data pages written to disk per second for all members during peak activity. Do not use the number of pages written to the group buffer pool; D must be a count of distinct pages. To determine this value, use the field QBSTCO from IFCID 0002 (the PAGES CASTOUT field of the buffer pool section of the Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS Statistics report.)

**R**
The average page residency time in the group buffer pool, in seconds. This value is application-dependent, but you can assume that the typical range is 30 to 180 seconds. If you have no information about residency time, use 120.

**General guideline:** Make R large enough so that other members can refresh an invalidated local copy of a changed page. When a changed page is written to the group buffer pool, it invalidates local copies of the page in other members. The changed page needs to remain resident in the group buffer pool long enough for other members to refresh the page.

**P**
The page size (4 KB, 8 KB, 16 KB, or 32 KB).

**VP**
The number of data pages defined for the virtual pool (the sum across all of the members).

0.2
The approximate size of a directory entry, in KB.

1.1
The additional storage needed for coupling facility control structures.

**Example:** Assume that you have a two-member data sharing group for which you have determined the following information:

- The degree of data sharing is very high (U=1).
- 500 disk writes per second occur across both members (D=500).
- No information exists about the page residency time (R=120).
- The page size is 4 KB (P=4).
- Member 1 is configured with a virtual pool of 80000 buffers.
- Member 2 is configured with a virtual pool of 40000 buffers (VP=120000).

The calculation is as follows:

\[
\begin{align*}
\text{Data_entries} & = 1 \times 500 \times 120 = 60000 \\
\text{Data(MB)} & = 60000 \times 4 / 1024 = 234 \text{ MB} \\
\text{Dir_entries} & = 60000 \times (1 \times 120000) = 1800000 \\
\text{Dir(MB)} & = 1.1 \times 180000 \times 0.2 / 1024 = 38 \text{ MB} \\
\text{GBP(MB)} & = 234 \text{ MB} + 38 \text{ MB} = 272 \text{ MB} \\
\text{RATIO} & = 180000 / 60000 = 3.0
\end{align*}
\]

The preceding calculation indicates that the group buffer pool should be defined with an INITSIZE of 272 MB. Use the ALTER GROUPBUFFERPOOL command to change RATIO to 3.

**General guideline:** For installation planning purposes, you should use the following general guideline as an initial estimate for the size of a Db2 group buffer pool for table spaces, indexes, or partitions that cache only changed data (GBPCACHE(CHANGED)):

Add the local buffer pool storage for this buffer pool number across all the members of the group. Then, multiply this amount by one of these workload factors:
Factor | Condition
---|---
10% | For light sharing with a low amount of updating activity
20% | For medium sharing with a moderate amount of update activity
40% | For a high amount of sharing with a lot of update activity

You can run a trace for IFCID 0002 to obtain an estimate of the amount of data sharing in your system. Calculate the "degree of data sharing" by dividing QBGLGG, the number of get pages for group buffer pool-dependent objects, by QBSTGET, the number of get pages. A value that is less than 25% is considered to be light data sharing, a value between 25% and 75% is medium data sharing, and a value greater than 75% is high data sharing.

Remember that the type of workload you run can influence the amount of storage you use. For example, if you have “hot spots” in which updates to a single page are frequent rather than spread throughout the table space, you might need less storage for caching.

**Example:** Assume that the total buffer pool storage for all the members of the group is 400 MB, and you expect a medium amount of read/write sharing in the environment. The following calculation is an estimate for the size of a group buffer pool when GBPCACHE(CHANGED):

\[
400 \text{ MB} \times 20\% = 80 \text{ MB}
\]

**Storage estimate for caching all data:**

For installation planning purposes, you can use an initial size estimate of a group buffer pool that caches read-only pages with changed pages (GBPCACHE(ALL)).

Calculate the sum of the local buffer pool storage for this buffer pool number (virtual only) across all members in the group. Then, multiply this amount by one of these workload factors:

<table>
<thead>
<tr>
<th>Factor</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>Less than half of the table spaces, indexes, or partitions specify GBPCACHE ALL</td>
</tr>
<tr>
<td>75%</td>
<td>Half of the table spaces, indexes, or partitions specify GBPCACHE ALL</td>
</tr>
<tr>
<td>100%</td>
<td>75% of more of the table spaces, indexes, or partitions specify GBPCACHE ALL for a high sharing environment</td>
</tr>
</tbody>
</table>

**Example:** Assume that the local buffer pool storage on all members of the group adds up to 200 MB. Half of the page sets that come into the pool are defined as GBPCACHE(ALL). The following calculation is an estimate for the size of a group buffer pool when GBPCACHE(ALL):

\[
200 \text{ MB} \times 75\% = 150 \text{ MB}
\]

**Storage estimate for caching no data:**

For planning purposes, you can use a formula to estimate the size of group buffer pools that cache no data.
The following is a formula for estimating storage for group buffer pools that cache no data (GBPCACHE(NO)):

\[
\begin{align*}
\text{Dir\_entries} &= U \times VP \\
\text{Dir\(\text{(MB)}\)} &= 1.1 \times \text{Dir\_entries} \times 0.2 / 1024 \\
\text{GBP\(\text{(MB)}\)} &= \text{Dir\(\text{(MB)}\)} \\
\text{RATIO} &= \text{n/a}
\end{align*}
\]

If the group buffer pool itself is defined with GBPCACHE(NO), the ratio is ignored.

In summary, the variables are:

- **U**: The estimated degree of data sharing.
- **P**: The page size (4 KB, 8 KB, 16 KB, or 32 KB).
- **VP**: The number of data pages defined for the virtual pool (the sum across all the members).

**Example:** Assume that you have a two-member data sharing group for which you have determined the following information:

- The degree of data sharing is very high (U=1).
- Member 1 is configured with a virtual pool of 80000 buffers.
- Member 2 is configured with a virtual pool of 40000 buffers (VP=120000).

The calculation is as follows:

\[
\begin{align*}
\text{Dir\_entries} &= 1 \times 120000 = 120000 \\
\text{Dir\(\text{(MB)}\)} &= 1.1 \times 120000 \times 0.2 / 1024 = 25 \text{ MB} \\
\text{GBP\(\text{(MB)}\)} &= 25 \text{ MB}
\end{align*}
\]

The preceding calculation indicates that the group buffer pool should be defined with an INITSIZE of 25 MB. Use the command ALTER GROUPBUFFERPOOL to change the GBPCACHE attribute to NO. If you put GBPCACHE NONE page sets in a GBPCACHE(YES) group buffer pool, the calculation becomes more complicated. You do this because the RATIO is observed and you are probably going to waste a lot of space on unneeded data entries.

**Related concepts:**

- **Storage estimate for group buffer pools that cache changed data**

**Storage estimate for caching LOB space maps (GBPCACHE SYSTEM):**

For planning purposes, you can use a formula to estimate storage for group buffer pools that cache LOB space map data.

The following is a formula for estimating storage for group buffer pools that cache LOB space map data:

\[
\begin{align*}
\text{Data\_entries} &= (U \times D / 10) \times R \\
\text{Data\(\text{(MB)}\)} &= \text{Data\_entries} \times P / 1024 \\
\text{Dir\_entries} &= \text{Data\_entries} + (U \times VP) \\
\text{Dir\(\text{(MB)}\)} &= 1.1 \times \text{Dir\_entries} \times 0.2 / 1024 \\
\text{GBP\(\text{(MB)}\)} &= \text{Data\(\text{(MB)}\)} + \text{Dir\(\text{(MB)}\)} \\
\text{RATIO} &= \text{MIN(Dir\_entries / Data\_entries, 255)}
\end{align*}
\]

In summary, the variables are:

- **U**: The estimated degree of data sharing.
- **D**: The number of data pages that are written to disk per second for all
members during peak activity. Do not use the number of pages that are written to the group buffer pool; D must be a count of distinct pages. To determine this value, use the field QBSTPWS from IFCID 0002 (the PAGES WRITTEN field of the buffer pool section of the Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS Statistics report).

10 An estimate of the LOB system pages that are written for every LOB data page.

P The page size (4 KB, 8 KB, 16 KB, or 32 KB).

R The average page residency time in the group buffer pool in seconds.

VP The number of data pages that are defined for the virtual pool (the sum across all the members).

Example: Assume that you have a two-member data sharing group for which you have determined the following information:

- The degree of data sharing is moderate (U=.7).
- 10 disk writes per second occur across both members during peak activity (D=10).
- The space map page is resident in the group buffer pool page for 120 seconds (R=120).
- The page size is 32 KB (P=32).
- Member 1 is configured with a virtual pool of 20000 buffers.
- Member 2 is configured with a virtual pool of 10000 buffers (VP=30000).

The calculation is as follows:

\[
\text{Data\_entries} = \left(\frac{.7 \times 10}{10}\right) \times 120 = 84
\]

\[
\text{Data(MB)} = \frac{84 \times 32}{1024} = 2.6 \text{ MB}
\]

\[
\text{Dir\_entries} = 84 + \left(\frac{.7 \times 30000}{1024}\right) = 21084
\]

\[
\text{Dir(MB)} = 1.1 \times 21084 \times 0.2 / 1024 = 4.5 \text{ MB}
\]

\[
\text{GBP(MB)} = 2.6 \text{ MB} + 4.5 \text{ MB} = 7.1 \text{ MB}
\]

\[
\text{RATIO} = \text{MIN}(21084 / 84, 255) = 251
\]

The preceding calculation indicates that the group buffer pool should be defined with an INITSIZE of 7.1 MB. The ratio is less than the maximum value, so use the command ALTER GROUPBUFFERPOOL to change the ratio to 251.

Related concepts:

- Storage estimate for group buffer pools that cache changed data

Related reference:

- ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)

PR/SM formulas for calculating sizes of group buffer pools:

You can use Processor Resource/System Manager (PR/SM™) formulas to calculate the sizes of group buffer pools.

You can also calculate group buffer pool sizes by using the coupling facility allocation formulas for cache structures found in zSeries Processor Resource/Systems Manager Planning Guide. The following table contains information that is used in those formulas. The size of cache structures in Db2 can vary greatly based on the amount of data for which inter-Db2 read/write interest exists at any given time. You probably need to monitor the use of the group buffer pools and adjust their sizes accordingly.
Table 61. Information for calculating cache structure sizes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Db2 Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSC</td>
<td>1</td>
<td>Maximum storage class</td>
</tr>
<tr>
<td>MCC</td>
<td>1024</td>
<td>Maximum castout class</td>
</tr>
<tr>
<td>MDAS</td>
<td>Dependent on page size being cached: 1 for 4 KB pages 2 for 8 KB pages 4 for 16 KB pages 8 for 32 KB pages</td>
<td>Maximum data area size</td>
</tr>
<tr>
<td>DAEX</td>
<td>4</td>
<td>Data area element characteristic</td>
</tr>
<tr>
<td>AAI</td>
<td>0</td>
<td>Adjunct assignment indicator</td>
</tr>
<tr>
<td>R_de</td>
<td>Set on ALTER GROUPBUFFERPOOL. Default is 5. See the table below for more information on determining this value.</td>
<td>The directory portion of the directory-to-data ratio</td>
</tr>
<tr>
<td>R_data</td>
<td>Set on ALTER GROUPBUFFERPOOL command. See the table below for more information on determining this value.</td>
<td>The data object portion of the target directory-to-data ratio</td>
</tr>
</tbody>
</table>

The following table contains information about the formulas used to determine \( R_{\text{data}} \) and \( R_{\text{de}} \). \( N \) is the RATIO entered on the ALTER GROUPBUFFERPOOL command. If \( N \) has a decimal point, all digits after the first decimal place are ignored. For example, 5.67 is treated as 5.6.

Table 62. Formulas for determining \( R_{\text{data}} \) and \( R_{\text{de}} \).

<table>
<thead>
<tr>
<th>Page Size</th>
<th>( N ) has no decimal point</th>
<th>( N ) has decimal point</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 KB ( R_{\text{de}} )</td>
<td>( N )</td>
<td>( N \times 10 )</td>
</tr>
<tr>
<td>4 KB ( R_{\text{data}} )</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>8 KB ( R_{\text{de}} )</td>
<td>( N )</td>
<td>( N \times 10 )</td>
</tr>
<tr>
<td>8 KB ( R_{\text{data}} )</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>16 KB ( R_{\text{de}} )</td>
<td>( N )</td>
<td>( N \times 10 )</td>
</tr>
<tr>
<td>8 KB ( R_{\text{data}} )</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>32 KB ( R_{\text{de}} )</td>
<td>( N )</td>
<td>( N \times 10 )</td>
</tr>
<tr>
<td>32 KB ( R_{\text{data}} )</td>
<td>8</td>
<td>80</td>
</tr>
</tbody>
</table>

Related reference:

ALTER BUFFERPOOL (Db2) (Db2 Commands)

Lock structure size
The coupling facility lock structure contains two parts: a lock entry table and a list of update locks.

The lock entry table is used to determine if inter-Db2 read/write interest exists on a particular hash class. Hash classes are resources that hash to a particular place in the lock table. The second part is a list of the update locks that are currently held (sometimes called a modify lock list or record list table). You can control the division
of the lock structure storage between these two components by using the IRLMPROC or by using an IRLM MODIFY command. If you do not specify how the structure is to be split, IRLM attempts to divide it with a 1:1 ratio between lock table entry (LTE) and record list entry (RLE) storage.

The total size of the lock structure must be large enough to limit hash contention, which prevents performance problems. The lock structure must also be large enough to prevent failures that result from a lack of record table storage to write a MODIFY entry (RLE). Proper specification for the number of LTEs can help avoid hash contention.

IRLM reserves 10% of the record table entries for “must complete” functions (such as rollback or commit processing), so that a shortage of storage does not cause a member to fail. However, if storage runs short in the record table, there can be an impact on availability (transactions are terminated), response time, and throughput.

The z/OS system can automatically alter the size of a lock structure in the coupling facility if it needs storage space. The ALLOWAUTOALT option in the CFRM policy specifies whether system-initiated automatic alters are to be allowed for a structure.

If no competing resource is using the coupling facility, consider specifying ALLOWAUTOALT(YES) to enable the lock structure to change as the workload grows. However, if you use this setting, other resources can force the IRLM lock structure size to decrease. This situation can cause an IRLM out-of-storage condition in the lock table and an abend.

If you need the lock structure to be stable, specify ALLOWAUTOALT(NO). If you do not specify ALLOWAUTOALT, the default value of NO is used.

Related concepts:

- Avoiding false contention (Db2 Data Sharing Planning and Administration)
- Auto Alter capabilities (Db2 Data Sharing Planning and Administration)
- Changing the size of the lock structure (Db2 Data Sharing Planning and Administration)
- Ways to monitor Db2 locking activity (Db2 Data Sharing Planning and Administration)

Lock entry size parameter:

The LOCK ENTRY SIZE parameter on installation panel DSNTIPJ determines the amount of space required for lock contention control information (that is, individual entries in the lock table).

The lock entry size and the number of lock table entries of the first IRLM to join the group determines the whole group’s storage size for the lock table and the lock table entry width. The default is 2 bytes, which is probably the size you want, unless you immediately create a data sharing group of seven or more members. By restricting each lock entry to 2 bytes, you maximize the amount of RLE space available from the define structure size.

Storage estimate for the lock structure:

For installation planning purposes, the initial size of the lock structure is based on how much updating you do.
**Recommendation:** If you do not specify a value for the LTE parameter in the IRLMPROC procedure, choose a value for INITSIZE that is a power of 2. This enables IRLM to allocate the coupling facility storage so that half is used for lock table entries and the remainder is used for record table entries. If a 1:1 split occurs and total size is not a power of 2, you might experience a severe shortage of space for the record table entries, resulting in Db2 or IRLM failures. Those failures occur because the number of lock table entries requested during connection to the group must be a power of 2. The record table is susceptible to storage shortages if the structure is too small or if the allocation of the lock table leaves too little storage for the record table.

When you specify a value for the LTE parameter in the IRLMPROC procedure, or when you issue the `MODIFY irlimproc SET,LTE=nnnn` command, you should monitor XES contention rates to determine the optimum value for your normal operating environment. If the contention rates appear to be too high, increase the LTE parameter value to the next power of 2. Remember, any increase in the size of the lock table causes a corresponding decrease in the record table, unless the structure size is also increased.

If you have little contention and want more storage available for record table entries, decrease the LTE parameter value by a power of two. Anytime the number of lock table entries is decreased, you should monitor contention rates for a period of time.

Because structure allocation is done during connection, any change that is made to the LTE parameter value does not take effect unless the lock structure is reallocated. You can reallocate the lock structure in one of the following ways:

- Rebuild the lock structure. This is the simpler method.
- Terminate the group, force deallocation of the lock structure, and restart the group.

The LTE parameter value of the first IRLM to connect to the newly allocated structure dictates the coupling facility structure sizes that are used by the group.

The following table shows initial size value recommendations.

<table>
<thead>
<tr>
<th>INITSIZE</th>
<th>SIZE</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 MB</td>
<td>64 MB</td>
<td>For light sharing with a low amount of updating, or for a single-member data sharing group</td>
</tr>
<tr>
<td>64 MB</td>
<td>128 MB</td>
<td>For medium sharing with a moderate amount of update activity</td>
</tr>
<tr>
<td>128 MB</td>
<td>256 MB</td>
<td>For a high amount of sharing with a lot of update activity</td>
</tr>
</tbody>
</table>

**SCA size**

The shared communications area (SCA) is a list structure in the coupling facility. Your storage estimate for your data sharing group should include an estimate of SCA size.

The SCA contains the following information:

- Member names
- BSDS names
• Database exception status conditions about object in the database
• Recovery information, such as log data set names and the list of indoubt XA transactions

You can use the coupling facility structure sizing tool (CFSizer) to help you calculate structure sizes. The following table shows how to estimate the size of the SCA. The SCA size can be specified in 1 KB increments.

Table 64. Estimating storage for the SCA

<table>
<thead>
<tr>
<th>Site size</th>
<th>Databases</th>
<th>Tables</th>
<th>INITSIZE</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>50</td>
<td>1000</td>
<td>16 MB</td>
<td>32 MB</td>
</tr>
<tr>
<td>Medium</td>
<td>200</td>
<td>4000</td>
<td>32 MB</td>
<td>64 MB</td>
</tr>
<tr>
<td>Large</td>
<td>400</td>
<td>10,000</td>
<td>64 MB</td>
<td>128 MB</td>
</tr>
<tr>
<td>Extra Large</td>
<td>600</td>
<td>50,000</td>
<td>128 MB</td>
<td>256 MB</td>
</tr>
</tbody>
</table>

Running out of space in the SCA can cause Db2 to fail. Because much of the space in the SCA is taken up with exception information, you can reclaim space by correcting database exception conditions. Each indoubt XID entry takes approximately 250 bytes. The additional SCA storage required to be reserved is dependent on the total number of database access threads (DBATs) that can be active for the data sharing group. If the data sharing group supports XA transactions, be sure to reserve additional space in the SCA for the indoubt transaction ID (XID) list. (DRDA XA protocol support enables Db2 servers to participate in WebSphere coordinated two-phase commit transactions that have multiple connections to more than one data source.) After a failure, indoubt XA transactions are resolved by WebSphere, or you can reclaim space by resolving indoubt XA transactions manually.

Related tasks:
- Recovering indoubt threads (Db2 Administration Guide)
- Determine the size of each coupling facility structure (z/OS MVS Setting Up a Sysplex)

How to change structure sizes

You can change the size of individual structures by changing the CFRM policy.

Then you should rebuild the structures by using the z/OS command SETXCF START,REBUILD,STRNAME=strname. Be aware that:

• The coupling facility size does not increase unless at least one Db2 subsystem is active when this command is issued.
• This command does not work on structures that are actively being duplexed.

z/OS attempts to reallocate a new instance of the structure in the same coupling facility if that coupling facility has enough storage space. If there is not enough space, z/OS looks at the preference list and uses the specified alternate coupling facility. After the space is allocated, Db2 rebuilds the information into the new structure. Any transactions that need the structure must wait until the rebuild is complete. It is best to plan a rebuild when other activity in the system is low.
Dynamic changes to structure sizes:

If the affected structure is allocated in a coupling facility with CFLEVEL greater than 0, you can dynamically increase the structure sizes up to a maximum limit specified on the CFRM policy by using the z/OS command SETXCF START,ALTER.

The advantages to this method are:

- Db2 can access the structures while a change is taking place.
- Less coupling facility storage is required because Db2 does not need to allocate enough space for a whole new structure. It dynamically adds or deletes storage from the existing structure.
- The command works on duplexed structures.

Related concepts:

Storage estimates for data sharing environments

Changing the size of the group buffer pool (Db2 Data Sharing Planning and Administration)

Problem: storage shortage in the SCA (Db2 Data Sharing Planning and Administration)

Related tasks:

Changing the lock structure size dynamically (Db2 Data Sharing Planning and Administration)

Related reference:

Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)

Automatic changes to structure sizes:

z/OS can automatically alter a coupling facility structure when the structure reaches an installation-defined or default percent-full threshold, as determined by structure full monitoring.

The alter process can change the size of the structure, reapporportion the objects within the structure, or both. This function works the same in data sharing environments as it does in non-data sharing environments. See Adding MVS systems to a sysplex for complete information about allowing structures to be altered automatically.

IRLM storage size

For data sharing, plan for additional IRLM storage to accommodate data sharing-specific locks called P-locks.

P-locks are held on open page sets and on database descriptors (DBDs), skeleton cursor tables (SKCTs), and skeleton package tables (SKPTs). Unlike transaction locks, storage for P-locks is held even when no transaction activity exists; therefore they consume storage even with no transaction activity.

Plan also for the extra storage that IRLM needs to build retained locks in case other members fail. The following table shows the variables that you need to take into account.
Table 65. Variables that are used to estimate additional IRLM storage

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>P-locks ¹</td>
<td>N = (MAX_OPEN_DATA_SETS × 500) [X = (N + (N \times .40))]</td>
</tr>
<tr>
<td>Y</td>
<td>Ability to hold update-retained locks for a failed member</td>
<td>Depends on the update intensity of the workload. Start with the following: [Y = .25(X)(X)]</td>
</tr>
</tbody>
</table>

Note:

1. The formula is based on the assumption that more than one P-lock might be held on a page set occasionally (such as for castout activity), and an estimate of about 40% for P-locks on the EDM pool objects and for short-lived page P-locks. If you know that the EDM pool has relatively few objects in it, you can use a lower percentage for that value. Estimate the maximum number of open data sets or use the value that is specified for the subsystem parameter DSMAX.

Related concepts:
- IRLM address space (IRLMPROC) storage requirements
- How Db2 determines the initial value of DSMAX (Db2 Performance)
- Physical locks in data sharing (Db2 Data Sharing Planning and Administration)

Related reference:
- DSMAX field (DSMAX subsystem parameter)

IRLM priority setting:

If IRLM priority is too low, storage might not be freed as quickly, and IRLM might run out of storage.

The IRLM address space priority, though it should remain high, should be lower than XCFAS.

Related tasks:
- Determining z/OS Workload Manager velocity goals (Db2 Performance)

IRLM storage use:

Use the z/OS command MODIFY irlmproc,STATUS,STOR to see how much storage IRLM is using.

Related reference:
- MODIFY irlmproc,STATUS (z/OS IRLM) (Db2 Commands)

Storage estimate for the EDM pool in a data sharing environment

If you are planning to use data sharing, you should consider additional storage requirements for the EDM pool.

General guideline

You can use the storage estimates for your capacity planning.
For data sharing, you might need to increase your storage estimate by about 10% because of the way Db2 cross-invalidates database descriptors (DBDs). This percentage is just an estimate; the actual amount of the increase depends on how often you create, drop, and alter objects in the data sharing group.

**Cross-invalidating items in the EDM pool**

Db2 does not have a backup EDM pool in the coupling facility for invalidating objects in the EDM pool (DBDs, cursor tables, and more) because these objects are modified less frequently than database data. So, there is one EDM pool for each member. When a DBD changes, Db2 uses XCF messages to notify other members which are also using that DBD that new transactions should use the new DBD, which is read into the EDM pool. Thus, it is possible that one transaction is using the new DBD while other, currently running transactions are still using the old DBD. In other words, more than one copy of a DBD can exist in the EDM pool.

**Reducing the storage impact**

For CREATE, ALTER, or DROP statements, the DBD is not modified until a COMMIT is issued. You can significantly reduce the number of EDM versions by issuing CREATE, ALTER, or DROP statements within a single COMMIT scope. However, the exclusive lock on the DBD is held until the COMMIT.

**Before you enable Db2 data sharing**

You must take certain actions to prepare your environment for Db2 data sharing.

**Connecting CICS to Db2 (optional)**

Before you enable data sharing, connect CICS to each member. The CICS attachment facility command, DSNC STRT, lets you override the subsystem name on startup.

**Connecting Db2 to IMS (optional)**

Before you enable data sharing, attach IMS to every member that runs IMS applications. IMS must include a separate member SSM for every member Db2.
About this task

Connecting Db2 to IMS requires coordination with your company’s group that supports IMS systems.

Procedure

To connect the IMS attachment facility:
1. Make Db2 load modules available to IMS
2. Define Db2 to IMS
3. Define new programs and transactions to IMS
4. Prepare IMS applications for Db2
5. Depending on your site, you might also need to:
   • Define Db2 plans for IMS applications
   • Generate a user language interface

Registering the command prefixes, member group attachment name, and subgroup attachment name

Before you enable data sharing, register the command prefix for a member, the group attachment name, and the subgroup attachment name for the group.

Procedure

To register the command prefixes, member group attachment name, and subgroup attachment name:

Use parameter library member IEFSSNxx. For example:

```
SUBSYS SUBNAME(ssname) INITRTN(DSNT3IN1)
   INITPARM('DSN3EPX,cmd-prefix<,scope<,group-attach<,subgroup-attach>>')
```

By default, the subsystem definition does not include the subgroup attachment name. If you specify a value for SUBGRP NAME on installation panel DSNTIPK, the installation CLIST appends the name to the subsystem definition.

**Recommendation:** When you register the 1- to 8-character command prefix in parameter library member IEFSSNxx and specify the scope of the prefix, choose a scope of started (S). This specification lets all z/OS systems in a single PARMLIB member IEFSSNxx use all z/OS systems in the Parallel Sysplex. It also simplifies the task of moving a member from one system to another; you can stop Db2 on one z/OS system and start it on another. You do not need to re-IPL the system.

Results

The following sample definitions are examples of what might appear in the shared parameter library SYS1.PARMLIB:

```
DB1A,DSN3INI,'DSN3EPX,-DB1A,S,DB0A'
DB2A,DSN3INI,'DSN3EPX,-DB2A,S,DB0A,SBG1'
DB3A,DSN3INI,'DSN3EPX,-DB3A,S,DB0A,SBG1'
DB4A,DSN3INI,'DSN3EPX,-DB4A,S,DB0A,SBG2'
```

**What to do next**

If you need to change the command prefix parameters, you must change entry IEFSSNxx and re-IPL the host system. For example, if you want to change the
command prefix scope from system-wide to Parallel Sysplex-wide, and you want
to register the prefix at Db2 startup, change the M in the entry to S before you
re-IPL.

If you want to use multiple-character command prefixes, make sure that your
automation programs can handle multiple-character prefixes in messages before
you change the prefixes.

Related concepts:
Group attachment names and subgroup attachment names

Related information:

Increasing the size of the BDSs
Data sharing causes additional records to be written to the BDSs for member
information.

About this task

Recommendations: To prevent the BDSs from expanding into secondary extents,
change the size of the primary space allocation to 1500 records. This change is
necessary only for members that have followed the recommended migration path
from a previous Db2 version without altering that record size. New installations
and member installations already do this for you.

Procedure

To increase the space allocation for the BDSs:
1. Rename existing BDSs.
2. Define larger BDSs with the original names.
3. Copy the renamed BDSs into the new BDSs.

Results

You can do this using access method services. To see the definition that is used for
the BDSs, see the installation job DSNTIJIN.

Increasing the size of the SYSLGRNX table space
The SYSLGRNX directory table space contains the relative byte address (RBA)
ranges showing when data sets are open for updating.

About this task

Because more members open and close data sets in a data sharing configuration,
this table space is likely to grow with the addition of each new member to the data
sharing group. And if you choose to copy indexes, this table space can grow even
more.

Consider increasing the frequency with which you remove rows from this table
space, or increasing the size. To see the definition used for SYSLGRNX, see
installation job DSNTIJIN.
**Procedure**

To increase the space allocation for SYSLGRNX, use access method services:

1. Stop the table space.
2. Rename the existing SYSLGRNX data set.
3. Define a larger SYSLGRNX data set with the original name.
4. Using only DSN1COPY, copy the contents of the renamed data set into the new SYSLGRNX data set.
5. Restart the table space.

**Database design recommendation**

You should partition table spaces as much as possible and use as much separation of I/O as possible.

*Define table spaces and indexes with GBPCACHE CHANGED:* It is unlikely that members that are processing a large query will repeatedly read the same pages, so those pages do not need to be cached in the group buffer pool. Define the relevant table spaces and indexes with GBPCACHE CHANGED (the default).

**Related tasks:**

- Partitioning for optimal parallel performance (Db2 Performance)

**Application design planning for data sharing**

Particular considerations apply to applications for use in Db2 data sharing environments.

**CACHE option of CREATE SEQUENCE statement**

The CACHE option of the CREATE SEQUENCE statement is a performance and tuning option that directs Db2 to preallocate a specified number of sequential values in memory.

Sequence objects provide unique, sequential, and numeric values to Db2 applications. Applications can use sequence numbers for a variety of purposes, including the avoidance of concurrency and performance problems that can result when applications generate their own sequence numbers. Unlike application-generated sequences, Db2 sequence objects allow multiple transactions to concurrently increment the sequence number, and Db2 guarantees that each sequential value is unique. Db2 does not wait for a transaction that has incremented a sequence to commit before allowing the sequence to be incremented by another transaction. No retained locks are held to prevent access to the sequence.

Every time a cache is allocated, the SYSIBM.SYSSEQUENCES table is updated and a forced log record is written for the update. Using ORDER or NO CACHE will result in a SYSIBM.SYSSEQUENCES table update and the associated forced log write every time a new value is generated by Db2. Specifying CACHE provides faster access to the sequence since values are assigned from the cache. It also reduces the number of SYSIBM.SYSSEQUENCES updates and the associated forced log records. The SYSIBM.SYSEQ table space is defined with the MAXROWS(1) attribute in order to minimize page P-lock contention in a data sharing environment as the SYSIBM.SYSSEQUENCES table gets updated.
Db2 always generates sequence numbers in order of request. However, when a sequence is shared across multiple members of a data sharing group, each Db2 member allocates its own cache of unique consecutive numbers for the sequence. Therefore, in situations where transactions from different members are requesting the next sequence number from the same sequence, values assigned for the sequence across multiple Db2 members may not be in strict numeric order.

**Example:** Assume that members DB2A and DB2B share a sequence named SEQ1 that starts with 1, increments by 1, and has cache = 20. If the transaction that is associated with DB2A makes the first request for a sequence number, DB2A allocates a cache of 20 values (from 1 to 20) and the value of 1 is provided to the application. If the transaction that is associated with DB2B makes the next request for a sequence number, DB2B allocates its own cache of 20 values (from 21 to 40) and the value of 21 is provided to the application. Assuming that sequence number requests continue to arrive from the transactions that are associated with members DB2A and DB2B in this manner (one from DB2A and then one from DB2B), the values assigned for that sequence are 1, 21, 2, 22, 3, 23, and so on. Although the numbers are in sequence within each Db2, the numbers assigned across multiple Db2 members are not in strict numeric sequence.

In a data sharing environment, using the ORDER or NO CACHE option ensures that the values assigned to a sequence which is shared by applications across multiple Db2 members are in strict numeric order. In a data sharing environment, if ORDER is specified, then NO CACHE is implicit even if CACHE n is specified.

In a non-data sharing environment, the numbers are always assigned in strict numerical order, even if NO ORDER is specified; so specifying ORDER or NO CACHE is not necessary.

**Important:** Specifying ORDER or NO CACHE in a data sharing environment, or NO CACHE in a non-data sharing environment, is not recommended because it will result in a SYSIBM.SYSEQUENCES table update and the associated forced log record for each value generated by Db2. This can lead to an excessive amount of log write I/Os, which can drastically reduce the log write bandwidth.

**Related reference:**
- CREATE SEQUENCE (Db2 SQL)

**Applications using CICSPlex System Manager**

CICSPlex® System Manager (CICSPlex SM) is a system-management tool that lets you manage several Customer Information Control System (CICS) systems as if they were one.

The dynamic transaction routing program that is supplied with CICSPlex SM balances the enterprise workload dynamically across the available application owning regions. CICSPlex SM lets you manage a variable workload without operator intervention and maintains consistent response times. It can do this because it routes transactions away from busy regions and from those that are failing or likely to fail, which improves throughput and conceals problems from users.
Storm-drain effect:

In some situations, your Db2 applications must be sensitive to a resource-unavailable condition.

For example, assume that a database is stopped for planned maintenance, and that the application receives SQLCODE -904 from the system and ends normally. If your application ignores the SQLCODE -904 message, CICSPlex SM might continue to route work to the stopped system because it appears to complete its work rapidly. This is sometimes called the storm-drain effect.

When both of the following conditions are true, the storm drain effect can occur:
  • The CICS attachment facility is down.
  • You are using INQUIRE EXITPROGRAM to avoid AEY9 abends.

Again, because there has not been an abend, it appears as if work completes rapidly at that subsystem.

How a CICS exit routine can avoid the storm-drain effect:

You can write a resource manager interface program exit routine, XRMIOUT, to avoid the storm drain effect caused by SQLCODE -904 (resource unavailable). This exit routine does not avoid the storm-drain problem caused by using INQUIRE EXITPROGRAM to avoid AEY9 abends.

About this task

GUP!

Using XRMIOUT, you can intercept the return from the resource manager. The exit routine can determine whether:
  • The resource manager is Db2.
  • SQLCODE -904 is in the SQL communication area (SQLCA).

If these conditions exist, abend the transaction instead of ending the transaction normally.

To determine if Db2 is the resource manager, compare 'DSNCSQL' with the value stored at the address that is included with the UEPTRUEN parameter that is passed to XRMIOUT, as shown in the following figure.
Procedure

To find the SQLCODE:

1. Find UEPTRUEP in the DFHUEPAR parameter list that is passed to XRMIOUT. UEPTRUEP contains the address of the DFHUEPAR parameter list that is passed to DSNCSQL.

2. Find UEPHMSA in DSNCSQL’s DFHUEPAR parameter list. UEPHMSA points to the register save area that contains the application’s registers.

3. Find register 1 in the register save area. Register 1 contains the address of the application parameters.
   
   The DSNXRDI macro maps the application parameters that are passed by the precompiler to Db2. The mapping macro is contained in the data set library prefix.SDSNMACS.

4. Find RDICODEP in the DSNXRDI structure. RDICODEP contains the address of the SQL communication area (SQLCA).

5. Find the SQLCODE.

Related concepts:

- SQL control statements for external SQL procedures (Db2 SQL)

Related reference:

- Exit XRMIOUT (CICS Transaction Server for z/OS)

A CICS enhancement that avoids the storm-drain effect:

The CICS Transaction Server for z/OS helps you avoid the storm-drain effect. With the CICS Transaction Server for z/OS, you do not need to use XRMIOUT to check for resource unavailable conditions.

CICS Transaction Server for z/OS also lets you benefit from the INQUIRE EXITPROGRAM command without causing the storm-drain effect.
Order-dependent transactions

In limited cases, one transaction, called an originating transaction, updates Db2 data with INSERT, UPDATE, or DELETE. Before completing phase 1 of the commit process, the originating transaction spawns a second transaction that is dependent on the updates of the first transaction. These types of transactions are referred to as order-dependent transactions.

Description of the problem

In some situations, a dependent transaction can encounter a “row not found” condition that does not occur in non-data sharing environments. Even in non-data sharing environments, dependent transactions need to tolerate the “row not found” condition in cases where the originating transactions roll back.

When the problem might occur

The “row not found” condition can occur if all of the following conditions are true:

- The originating transaction spawns a dependent transaction before it completes phase 1 of the commit process.
- The dependent transaction runs on a different member than the member on which the originating transaction runs.
  Because each member has its own local buffer pools, the uncommitted buffered pages that are updated by the originating transaction are not immediately visible to the dependent transaction.
- The dependent transaction attempts to access a row that was updated by the originating transaction.
- The dependent transaction is not bound with an isolation level of repeatable read.

Preventing the problem

To prevent these periodic “row not found” situations, consider using the IMMEDWRITE(YES) option of the BIND and REBIND commands for plans or packages that spawn dependent transactions that might run on other members. You can also use the IMMEDWRI subsystem parameter. IMMEDWRITE(YES) means that Db2 writes the page as soon as the buffer update completes. Db2 writes the data to one of the following structures or devices:

- The group buffer pool
- Disk storage for GBPCACHE NO group buffer pools
- Disk storage for GBPCACHE NONE page sets
- Disk storage for GBPCACHE SYSTEM page sets

The IMMEDWRI parameter (IMMEDIATE WRITE field on installation panel DSNTIP8) can override the value of the IMMEDWRITE bind option on a data sharing member. The following table illustrates the implied hierarchy when using the IMMEDWRI subsystem parameter and the IMMEDWRITE option of the BIND and REBIND commands.
Table 66. The implied hierarchy of the immediate write option

<table>
<thead>
<tr>
<th>IMMEDWRITE bind option</th>
<th>IMMEDWRI subsystem parameter</th>
<th>Value at run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Note: YES always has precedence whether it is defined by the subsystem parameter or the bind option. Updated pages that are group buffer pool-dependent are written at, or before, phase 1 of the commit process.

The following alternatives can help solve the order-dependent transaction problem:

- Ensure that the originating transaction does not schedule the dependent transaction until the originating transaction has completed phase 1 of commit.
- Run the dependent transaction with an isolation level of repeatable read.
- If the dependent transaction is currently running with an isolation level of cursor stability AND CURRENTDATA(NO), changing it to use CURRENTDATA(YES) can sometimes solve the problem.
- Add statement retry logic to handle the return of a “row not found” condition.
- Run the dependent transaction on the same member as the originating transaction.

Related concepts:

[The scope and uniqueness of Db2 subsystem parameters in data sharing]

Strategy for creating a Db2 data sharing group

When you create a data sharing group, add one member at a time. You can install a new Db2 for the first member (the originating member), or you can enable an existing Db2 subsystem as the originating member.

Before you begin

Familiarize yourself with the information in [Planning for Db2 data sharing]

About this task

Enabling Db2 data sharing is a big step. Carefully plan your strategy because data sharing is difficult to disable after it is enabled. Disabling should only be considered if your long-term plans are to disable data sharing. During migration, test other major new capabilities in the new release on a single system before enabling data sharing, and then build and try a test data sharing group. When you are ready to begin a move to production, you must avoid having to fall back to the previous release.

Tip: Db2 target library and distribution library data sets can be shared among all members of a data sharing group. Sharing these libraries reduces the effort of installing and maintaining the different members of the data sharing group. Sharing libraries can also help ensure that members are using the same exit routines. Also, sharing libraries simplifies the tasks of defining a new member to a data sharing group. The member installation process supports sharing of the libraries.
Procedure

Follow this general strategy for creating a data sharing group:

1. Build and try a test data sharing group. Here is one approach to testing your data sharing group:
   a. Install Db2 12 as a single system, and test it with dummy data, or copies of production data, in order to test some of the new capabilities. Run old applications and begin new application development work. In the meantime, prepare the hardware and define the coupling facility structures to enable data sharing.
   b. Enable data sharing on the test system. You have a single-system data sharing group at this point, and this can help you find any initial configuration problems. Make sure that old applications work in this environment.
   c. Install additional members in the test group. Run applications from different members in the group to fully exercise the group buffer pools and cross-system locking.

2. Move to production. When you are ready to move to production complete the following steps:
   a. If your originating member is not already migrated to Db2 12, migrate it from Db2 11 to Db2 12.

   **Recommendation**: Wait until your system is stable before enabling data sharing. You can enable data sharing on your existing release and then migrate to Db2 12, but it is not recommended, particularly if you plan to migrate soon. If you are in the process of migrating, wait until after the activation of function level 500 or higher before you move to the data sharing mode.
   b. Verify that function level 500 or higher is activated and that your Db2 system is stabilized. Once the system is stabilized, you can begin using new capabilities.
   c. Tune applications to contain the level of locking and lock contention rates. To reduce the effects of locking contention in a data sharing environment, first control locking costs in a non-data sharing environment. This gives you a baseline from which to perform further tuning after the move to data sharing.
   d. Enable data sharing on the originating member, and run applications on this one-member data sharing group.
   e. Install additional members as needed.

Related concepts:

- Introduction to Db2 data sharing (Db2 Data Sharing Planning and Administration)
- Database administration for data sharing (Db2 Data Sharing Planning and Administration)
- Concurrency and locks in data sharing environments (Db2 Data Sharing Planning and Administration)

Related tasks:

- Controlling resource usage (Db2 Performance)
Subsystem parameters and application defaults in a data sharing environment

In a data sharing environment, you must consider the scope of subsystem parameters before setting the values for each member. Also consider the subsystem parameter load module and the application defaults load module that you will specify for each member.

Specifying a member's subsystem parameter load module

Every member of a data sharing group must have its own unique load module for subsystem parameters (sometimes called DSNZPARM in a non-data sharing environment).

About this task

The load module for subsystem parameters is built by job DSNTIJUZ and stored in the prefix.SDSNEXIT target library. Every member must use a different name for its parameter load module because the prefix.SDSNEXIT target library can be shared among all members of the data sharing group. The installation process requires that you provide the name of the load module for a member.

Recommendation: Name each member's load module using the convention DSNZPxxx, where xxx includes the number in the member name and the group identifier. For example, DB1A's subsystem parameter load module could be named DSNZP01A.

The subsystem parameter load module name for a member is an optional parameter on the EXEC statement in the JCL procedure that is used to start the ssnmMSTR address space. This optional parameter provides support for an operator (or automated operations) not specifying the subsystem parameter load module name when starting a member.

Procedure

To specify the subsystem parameter load module for a member:

Specify the load module name in the EXEC statement of the ssnmMSTR JCL procedure in SYS1.PROCLIB as follows:

```
//IEFPROC EXEC PGM=DSNYASCP,PARM='ZPARM(DSNZPxxx)',...
```

Related tasks:

Tailoring Db2 12 installation and migration jobs with the CLIST

The scope and uniqueness of Db2 subsystem parameters in data sharing

Although various parameters affect the operation of only a single member, some parameters must have the same value on all the members. For example, each catalog alias name must be the same. Other parameter values must be different on each member. For example, each member specifies a different BSDS name.

The values of most parameters do not need to be unique. Recommendations are offered for some of these parameters. In the following tables, the parameter is indicated by the name of the corresponding installation panel field. However, some of the parameters do not reside in the DSNZPxxx load module.
Parameter values that must be different on each member

The list below shows the values of the installation panel fields that must be different for every member of a data sharing group. You specify values for these fields when a member is installed, migrated, or updated.

- **Active Logs**: COPY 1 PREFIX
- **Active Logs**: COPY 2 PREFIX
- **Archive Logs**: COPY 1 PREFIX (ARCPFX1 parameter)
- **Archive Logs**: COPY 2 PREFIX (ARCPFX2 parameter)
- **Bootstrap Data Sets**: COPY 1 NAME
- **Bootstrap Data Sets**: COPY 2 NAME
- **COMMAND PREFIX**
- **Db2 NETWORK LUNAME**
- **MEMBER IDENTIFIER**
- **MEMBER NAME**
- **PARAMETER MODULE**
- **PROC NAME**
- **RESYNC PORT**
- **SUBSYSTEM NAME**
- **SUBSYSTEM SEQUENCE**
- **SUBSYSTEM NAME (IRLM)**
- **WORK FILE DB**

Parameter values that must be the same on every member

The list below shows the installation panel fields that must have the same values for every member of the data sharing group.

- **APPL COMPAT LEVEL**
- **AUTH EXIT CACHE REFRESH**
- **AUTH EXIT CHECK**
- **CATALOG ALIAS**
- **COMPRESS_SPT01 parameter in macro DSN6SPRM**
- **DEL CF STRUCTS**
- **DBADM CREATE AUTH**
- **Db2 LOCATION NAME**
- **DESCRIBE FOR STATIC**
- **DIRECTORY AND CATALOG DATA**
- **DIRECTORY AND CATALOG INDEXES**
- **DRDA PORT**
- **EVALUATE UNCOMMITTED**
- **EXTENDED SECURITY**
- **GROUP ATTACH**
- **GROUP NAME**
- **INDEX SPACE ALLOCATION**
- **IRLM XCF GROUP NAME**
- **INSTALL DD CONTROL SUPT**
- **LIKE_BLANK_INSIGNIFICANT in macro DSN6SPRM**
Other recommendations

The table below shows the installation panel fields that can have the same or different values for members of a data sharing group.

Table 67. Recommended values for installation panel fields

<table>
<thead>
<tr>
<th>Installation panel field (parameter) name</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEALLOC PERIOD</td>
<td>Do not archive to tape. If you must, however, you should specify 0 for this parameter unless you intend to run all RECOVER jobs from the same member. Specifying a deallocation delay means that the tape is not available to any other member until the deallocation time expires.</td>
</tr>
<tr>
<td>DEFAULT 4-KB BUFFER POOL FOR USER DATA</td>
<td>For consistent results when creating objects on different members, make this default the same on all members.</td>
</tr>
<tr>
<td>DEFAULT 8-KB BUFFER POOL FOR USER DATA</td>
<td></td>
</tr>
<tr>
<td>DEFAULT 16-KB BUFFER POOL FOR USER DATA</td>
<td></td>
</tr>
<tr>
<td>DEFAULT 32-KB BUFFER POOL FOR USER DATA</td>
<td></td>
</tr>
<tr>
<td>DEFAULT BUFFER POOL FOR USER LOB DATA</td>
<td></td>
</tr>
<tr>
<td>DEFAULT BUFFER POOL FOR USER XML DATA</td>
<td></td>
</tr>
</tbody>
</table>

- MAX UTILS PARALLELISM
- MINIMUM DIVIDE SCALE
- OBJECT CREATE FORMAT
- PAD INDEXES BY DEFAULT
- PERCENT FREE FOR UPDATE
- REAL TIME STATS
- REVOKE DEP PRIV
- SECURE PORT
- SITE TYPE
- SKIP UNCOMM INSERTS
- STATISTICS FEEDBACK
- STATISTICS HISTORY
- STATISTICS ROLLUP
- SYSTEM ADMIN 1
- SYSTEM ADMIN 2
- SYSTEM OPERATOR 1
- SYSTEM OPERATOR 2
- SECURITY ADMIN 1
- SEC ADMIN 1 TYPE
- SECURITY ADMIN 2
- SEC ADMIN 2 TYPE
- SEPARATE SECURITY
- TABLE SPACE ALLOCATION
- TEMPLATE TIME
- TCP/IP ALREADY VERIFIED
- TRACKER SITE
- UNICODE IFCIDS
- UTILITY OBJECT CONVERSION
- VARY DS CONTROL INTERVAL

Chapter 9. Installing, migrating, and enabling Db2 data sharing
Table 67. Recommended values for installation panel fields  (continued)

<table>
<thead>
<tr>
<th>Installation panel field (parameter) name</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT BUFFER POOL FOR USER INDEXES</td>
<td>For consistent results when creating objects on different members, make this default the same on all members.</td>
</tr>
<tr>
<td>DEVICE TYPE 1 DEVICE TYPE 2</td>
<td>Archive the first copy of the log to disk.</td>
</tr>
<tr>
<td>DISABLE EDM RTS</td>
<td>For consistent results, make the value of this subsystem parameter the same on all members.</td>
</tr>
<tr>
<td>IMMEDIATE WRITE (IMMEDWR parameter)</td>
<td>This option should have the same value (either YES or NO) for all members of the group. Otherwise, the parameter, as it is set on the member that owns the archive log data set, determines which copy is used.</td>
</tr>
<tr>
<td>READ COPY2 ARCHIVE</td>
<td>This option should have the same value (either YES or NO) for all members of the group. Otherwise, the parameter, as it is set on the member that owns the archive log data set, determines which copy is used.</td>
</tr>
<tr>
<td>READ TAPE UNITS</td>
<td>Do not archive to tape. If you must, however, it is vital that you have enough tape units allocated to the member doing the recovery to merge the archive logs from all members in the group that have updated the object being recovered. Thus, if there are eight members in the group, make sure you specify at least eight on this panel for each member.</td>
</tr>
<tr>
<td>RECORDING MAX</td>
<td>All members should use the maximum value of 10000, making it easier to transfer a workload from one member to another.</td>
</tr>
<tr>
<td>RETAINED LOCK TIMEOUT</td>
<td>If you have automatic restart, or some other restart automation that quickly restarts failed members, choose a non-zero value for this parameter.</td>
</tr>
<tr>
<td>START IRLM CTRACE</td>
<td>Diagnostic traces should be activated for all members of the group. The negligible performance overhead is outweighed by the benefit of easier problem resolution. Only those systems that have reached the limits of processor capacity should consider not automatically activating these traces.</td>
</tr>
</tbody>
</table>

**Related concepts:**
- Order-dependent transactions
- The impact of archiving logs in a data sharing group (Db2 Data Sharing Planning and Administration)

**Related reference:**
- Directory of subsystem parameters, panel fields, and application default values
- Subsystem parameters that are not on installation panels
- BIND and REBIND options for packages, plans, and services (Db2 Commands)

**Updating subsystem parameters for a member**

No group-wide subsystem parameters can be updated; however, you can update the subsystem parameters for an individual member.

** Procedure **

To update subsystem parameters for a member of a data sharing group:

Specify the following installation option on the DSNTIPA1 installation panel:

```
INSTALL TYPE => UPDATE
```

This option updates parameters for the member.
Application defaults parameters

An application defaults load module, dsnhdecp, is generated as part of installing or migrating Db2. In a data sharing environment, this module is generated only for the first member that is installed or migrated.

If you provide a separate dsnhdecp module for each member, use the same dsnhdecp settings (except for the SSID parameter) for all members of the data sharing group. If you provide a separate dsnhdecp module for each member, after you activate a new function level, identify the jobs that you use to maintain those dsnhdecp modules. You must modify those jobs to specify SQLLEVEL=V12R1M500 or higher, and then run those jobs.

Installing a new data sharing group

Although you can install and immediately enable Db2 data sharing on a new Db2 12 member, you should do so only in low-risk situations. The recommended approach is to migrate to, or install, a Db2 12 member, use it for a while, and then enable data sharing.

Before you begin

Familiarize yourself with the following information:

- Planning for Db2 data sharing
- Strategy for creating a Db2 data sharing group

About this task

If you decide to install and immediately enable data sharing on a new Db2 12 member, that member becomes the originating member of the data sharing group. This member's Db2 catalog is used as the Db2 catalog for the entire data sharing group.

Procedure

To install a new data sharing group:

1. Run the installation CLIST as described in Tailoring Db2 12 installation and migration jobs with the CLIST. Then on installation panel DSNTIPA1, specify the following values:
   INSTALL TYPE ===> INSTALL
   DATA SHARING ===> YES
2. On installation panel DSNTIP0A, specify 1 to indicate the group data sharing function.
3. On installation panel DSNTIPK, specify:
   GROUP NAME ===> group name
   MEMBER NAME ===> originating member name
Verify that the originating member name is unique within the z/OS Parallel Sysplex. Installation job DSNTIJMV edits the ssnmMSTR startup procedure with the group name and member name you specify here.

4. Complete the installation panels and specify all of the parameters.

5. Complete the installation of Db2 by following the installation steps in Chapter 5, “Installing Db2,” on page 413.

6. Run the installation verification procedures (IVP).

Related concepts:
- Subsystem parameters and application defaults in a data sharing environment
- The scope and uniqueness of Db2 subsystem parameters in data sharing

Related tasks:
- Naming in Db2 data sharing
- Enabling Db2 data sharing

Related reference:
- DATA SHARING field (DSHARE subsystem parameter)

Related information:

---

### Enabling Db2 data sharing

The enabling process allows an existing Db2 subsystem to be the originating member of a data sharing group; it does not allow you to change the subsystem name.

**Before you begin**

Familiarize yourself with the following information:
- Planning for Db2 data sharing
- Strategy for creating a Db2 data sharing group

**About this task**

**Important:** If you use a product that uses IFCID 0306 to read log data, consult the documentation for that product for information about disabling and re-enabling data sharing.

**Tip:** You can use a variation of this process to rename data sharing members. For more information, see [Renaming data sharing members](#).

**Important:** If the RBA for a non data sharing Db2 subsystem reaches or exceeds the hard or soft limit, you must either convert all table spaces and indexes to the 10-byte format or use the RBA reset procedure. Enabling data sharing is not sufficient to resolve the problem.

**Procedure**

To enable Db2 data sharing, complete the following tasks:

**Related concepts:**
- The scope and uniqueness of Db2 subsystem parameters in data sharing

**Related tasks:**
- Installing Db2
Migration step 9: Stop Db2 11 activity

Running the CLIST for enabling data sharing

The process of enabling data sharing involves running some of the installation jobs. Before you run the installation jobs, tailor them to your environment by running the installation CLIST.

About this task

The enabling CLIST tailors the following jobs:

```
DSNTEJ0  DSNTEJ3M  DSNTEJ6W  DSNTEJ78
DSNTEJ1  DSNTEJ3P  DSNTEJ6Z  DSNTESA
DSNTEJ1L DSNTEJ4C  DSNTEJ61  DSNTESC
DSNTEJ1P DSNTEJ4P  DSNTEJ62  DSNTESD
DSNTEJ1S DSNTEJ5A  DSNTEJ63  DSNTESE
DSNTEJ1T DSNTEJ5C  DSNTEJ64  DSNTESH
DSNTEJ1U DSNTEJ5P  DSNTEJ65  DSNTIJDE
DSNTEJ2A DSNTEJ6   DSNTEJ66  DSNTIJFT
DSNTEJ2C DSNTEJ6D  DSNTEJ67  DSNTIJGF
DSNTEJ2D DSNTEJ6O  DSNTEJ7   DSNTIJIN
DSNTEJ2E DSNTEJ6P  DSNTEJ71  DSNTIJMV
DSNTEJ2F DSNTEJ6R  DSNTEJ73  DSNTIJTM
DSNTEJ2H DSNTEJ6S  DSNTEJ75  DSNTIJUZ
DSNTEJ2P DSNTEJ6T  DSNTEJ76  DSNTIJVC
DSNTEJ2U DSNTEJ6U  DSNTEJ77  DSN8ES1
DSNTEJ3C DSNTEJ6V  DSNTEJ78  DSN8ES2
```

The enabling CLIST also edits the DB2I CLISTs.

Procedure

To tailor the installation jobs for enabling data sharing:

1. Invoke the installation CLIST as described in Tailoring Db2 12 installation and migration jobs with the CLIST.

2. On installation panel DSNTIPA1 specify the following field values.

```
INSTALL TYPE  ===> INSTALL
DATA SHARING  ===> YES

INPUT MEMBER NAME  ===> input member for originating member
OUTPUT MEMBER NAME  ===> new output member
```

3. On installation panel DSNTIP0A, specify option 3 to enable data sharing.

4. On installation panel DSNTIPK, specify:

```
GROUP NAME  ===> group name
MEMBER NAME  ===> originating member name
```

Verify that the originating member name is unique within the z/OS Parallel Sysplex. Installation job DSNIJMV edits the ssnnMSTR startup procedure with the group name and member name that you specify here.

5. Complete the remaining installation panels, specifying parameters according to the Db2 subsystem parameters guidelines.

Related concepts:

Chapter 9. Installing, migrating, and enabling Db2 data sharing
Stopping Db2 activity

Before running any installation jobs to enable data sharing, ensure that all work is stopped on the subsystem that you chose to be the originating member.

Procedure

To stop work on Db2:
1. Issue the following command, where -DSN1 is the subsystem command prefix that is defined for Db2. You must have system administrator (SYSADM) or system operator (SYSOPR) authority to issue this command.

   -DSN1 STOP DB2 MODE(QUIESCE)

   The QUIESCE keyword allows Db2 to complete the processing of programs that are currently executing. This might require some processing time.

2. Issue the following command to allow only the system administrators and system operators that were defined during installation to access Db2:

   -DSN1 START DB2 ACCESS(MAINT)

   If Db2 does not start properly, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. For example, if the bootstrap data set (BSDS) does not match the subsystem parameter values, ensure that the correct jobs were run for DSNTIJUZ. Ensure that you started Db2 with the correct subsystem parameter option.

3. Follow these steps to ensure that all Db2 11 work is stopped:
   • Make sure no units of recovery remain. Issue the following command:

     -DSN1 DISPLAY THREAD(*) TYPE(*)

     Then use -DSN1 RECOVER INDOUBT for any indoubt threads.
   • Ensure that all utilities are completed or terminated. To ensure that no utility work remains, issue the following command:

     -DSN1 DISPLAY UTILITY(*)
After you have determined the utilities that are running, you can let them complete processing or you can terminate the utility. To stop all utilities, issue the following command:

-DSN1 TERM UTILITY(*)

- Ensure that no table spaces and index spaces in the Db2 directory (DSNDB01) or the Db2 catalog (DSNDB06) have write error ranges or deferred restart states. To determine existing restrictions, issue the following commands:

-DSN1 DISPLAY DATABASE(DSNDB01) SPACENAM(*) RESTRICT
-DSN1 DISPLAY DATABASE(DSNDB06) SPACENAM(*) RESTRICT

You must have system administrator or system operator authority (specified on installation panel DSNTIP1) to issue this command.

Recover any table spaces and index spaces that have write error ranges or deferred restart states by using the RECOVER utility.

4. Stop Db2 by issuing the following command:

-DSN1 STOP DB2 MODE(QUIESCE)

A user with installation system administrator (installation SYSADM) or installation system operator (installation SYSOPR) authority must enter this command.

Related reference:

- STOP DB2 (Db2) (Db2 Commands)
- DISPLAY THREAD (Db2) (Db2 Commands)
- DISPLAY UTILITY (Db2) (Db2 Commands)
- TERM UTILITY (Db2) (Db2 Commands)
- DISPLAY DATABASE (Db2) (Db2 Commands)

Modify the Db2 definition to IRLM and MVS for data sharing: DSNTIJMV

You must modify the definition of Db2 for data sharing in the system parameter libraries of the target z/OS system.

About this task

Job DSNTIJMV demonstrates how to update the SYS1.PARMLIB members to identify Db2 to IRLM and MVS. Although you might edit and run this job, the
more common approach is to examine the changes that DSNTIJMV makes and use an editor, such as ISPF/PDF, to make the required updated directly to the SYS1.PARMLIB members.

Because DSNTIJMV cannot anticipate all the necessary updates to identify Db2 to MVS and IRLM in your z/OS environment, additional updates might be required for your site. You can complete these updates either by making the updates directly in SYS1.PARMLIB members, or by editing and running the DSNTIJMV job.

Do not overwrite the entries for Db2 and IRLM in the program properties table (PPT). Note that PPT entries should no longer be defined in SYS1.PARMLIB(SCHEDxx). Default entries are supplied in the SYS1.LINKLIB(IEFSDPPT) load module, which is provided by z/OS. For details of the PPT entries, see Program properties table (PPT) (MVS Initialization and Tuning Reference).

**Procedure**

To modify the definition of Db2 for data sharing in the system parameter libraries of the target z/OS system:

1. Examine the DSNTIJMV job and study the updates that it makes.
2. Update the IEFSSNxx member of SYS1.PARMLIB by editing it directly, or by tailoring and running job DSNTIJMV. The IEFSSNxx member contains an entry for every z/OS subsystem. Db2 adds one entry for Db2 and two entries for IRLM in the list. The second IRLM entry, whose subsystem name is JRLM, simplifies adding a second IRLM to your system if the first is damaged. You must provide your own procedure to add JRLM. Unique names must be used for each entry.

z/OS provides subsystem entries for Db2 and IRLM in IEFSSN00. Examine these entries to determine whether they are appropriate for your needs. Make sure that a subsystem name appears only once in the subsystem name list.

Place the SMS line as the first line in an IEFSSNxx member. Place the primary system’s record (JES2 or JES3) record next. Other products might affect the position of the entries during system initialization. The Db2 line should come after SMS, the JES subsystem, and other vendor products.

The Db2 entry has the following format:

```
SUBSYS SUBNAME(ssname) INITRTN(DSN3INI)
  INITPARM('DSN3EPX,cmd-prefix<,scope<,group-attach<<,subgroup-attach>>>')

ssname
  The Db2 subsystem name.

DSN3INI
  The name of the Db2 load module that z/OS invokes during master scheduler initialization. This module must be located in a link list data set (or in SYS1.LINKLIB).

DSN3EPX
  The name of the Db2 load module that responds to Db2 requests that are received from the z/OS subsystem interface. (Db2 can be active or inactive when the requests are received.) This module must be located in a link list data set (or in SYS1.LINKLIB).

cmd-prefix
  The one- to eight-character command prefix.
The first character of the command prefix must be one of the following characters: @ # . / ) * + - = < > | & ! ; % _ ? : " . The remaining characters of the command prefix must be one of the above characters, A-Z, or 0-9. Do not use the JES2 backspace character or command prefix character. The default is the hyphen (-).

Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or z/OS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix beginning from the first character. For example, you cannot assign '-' to one subsystem and '-Db2A' to another. Similarly, you also cannot assign '?DB2' to one subsystem and '?DB2A' to another. You can assign '-DB2A' and '-DB2B' to different Db2 subsystems.

**scope**

The one-character scope for the command prefix. Db2 registers its command prefix with the operating system. When this is done, the scope of the command prefix is controlled by the value you choose:

- **S** Started; and the prefix is registered with Sysplex scope at Db2 startup instead of during z/OS IPL. This is the default.
  
  **Recommendation:** Choose S, which allows you to have a single IEFSSNxx PARMLIB member that all z/OS systems in the Sysplex can use. Choosing S also simplifies the task of moving a Db2 from one system to another; you can stop Db2 on one z/OS and start it up on another without having to re-IPL the system.

- **M** z/OS system scope; the prefix is registered during z/OS IPL.

- **X** Sysplex scope; the prefix is registered during z/OS IPL. As a result, this Db2 cannot be restarted on another z/OS without changing the definitions and re-IPLing both z/OS systems.

**group-attach**

The group attachment name, which is used for data sharing. You can specify this on installation panel DSNTIPK.

**subgroup-attach**

is the name of the subgroup attachment, which is used in data sharing environments to provide more control over how TSO, CAF, RRSAF, JDBC, ODBC, and Db2 utilities connections find active Db2 subsystems. You can specify this value by entering a subgroup name in the SUBGRP ATTACH field on installation panel DSNTIPK.

**Update address space startup procedures for Db2 data sharing: DSNTIJMA**

When you enable Db2 data sharing, you must modify the address space startup procedures for the ssnimMSTR and ssnimIRLM address spaces in the system procedure library for the target z/OS system.

**Before you begin**

The authorization ID that runs the jobs must have authority to update the target system PROCLIB. You can enter that library name on installation panel DSNTIPW in the SYSTEM PROCEDURES field. You might want to use a procedure library other than SYS1.PROCLIB for the procedures.
Examine the updates from the jobs in this step carefully before you run them. Also, you might have additional procedures of your own to rename.

**About this task**

Because different sites have different requirements, the jobs in the following steps cannot anticipate all the necessary updates to the target system procedure library in your z/OS environment. You can complete these updates either by making the updates directly in the target system procedure library, or by editing the job in the following steps.

**Procedure**

To modify the address space startup procedures for Db2 data sharing:

Tailor and run job DSNTIJMA. Job DSNTIJMA defines the startup procedures in the target system procedure library for the following Db2 address spaces, where `ssnm` is the subsystem name:

- `ssnmMSTR` for Db2 system services
- `IRLMPROC`, or a user-defined procedure for IRLM

### Altering the current active log data sets: DSNTIJIN

When job DSNTIJIN is tailored for enabling a Db2 subsystem to become the originating member of a new data sharing group, it does not define any data sets. It only alters the member’s active log data sets to use SHAREOPTIONS (2 3).

**Procedure**

To alter the active log data sets:

Run job DSNTIJIN. If DSNTIJIN runs successfully, it produces return codes of 0. Check any VSAM messages carefully.

**Related concepts:**

- Allocating and extending data sets (Db2 Performance)

### Defining Db2 initialization parameters for enabling data sharing

For Db2 12 installation, you must define initialization parameters in the subsystem parameter module (`DSNZPxxx`) and application defaults load module (`dsnhdecp`), define the offline message generator CCSID (`DSNHMCID`), and update DDF related information in the BSDS. For migration to Db2 12, you must define initialization parameters in the subsystem parameter module (`DSNZPxxx`) and application defaults load module (`dsnhdecp`), and define the offline message generator CCSID (`DSNHMCID`).

**About this task**

The subsystem parameter module (`DSNZPxxx`) and application defaults module (`dsnhdecp`) names are specified in the following DSNITPO3 installation panel fields:

- `PARAMETER MODULE` field
- `APPL DEFAULTS` field
Procedure

To define the Db2 12 initialization parameters, complete the following steps:

1. Run job DSNTIJUZ. Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

You might need to make the following adjustments before running the job:

   • If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLOAD, you can move the SYSLMOD output to that library.
   • If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
   • If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This action ensures that members DSNFSYSP, DSNJARVP, DSNJLOGP, DSNTSPRM, DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

     SET BOUNDARY (GLOBAL).
     LIST DDEF ( SMPTLIB )

     Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions. For more information, see Job DSNTIJUZ: define the Db2 data-only subsystem parameter module

2. Run job DSNTIJUA. Job DSNTIJUA defines the Db2 data-only application defaults module.

You might need to make the following adjustments before running the job:

   • If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLOAD, you can move the SYSLMOD output to that library.
   • If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUA to correct the data set names.
   • If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUA so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZQ. This action ensures that member DSNARIB is linked with the he application defaults module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

     SET BOUNDARY (GLOBAL).
     LIST DDEF ( SMPTLIB )

     Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

For more information, see Job DSNTIJUA: define data-only application defaults module

3. Run job DSNTIJUM. Job DSNTIJUM defines the Db2 data-only offline message generator CCSID (DSNHMCID) module which is needed for message
conversion by Db2 applications and utilities. For more information, see Job DSNFIJJUM: define Db2 data-only offline message generator CCSID (DSNHMCID) module.

4. Run job DSNFIJUL. Job DSNFIJUL runs the Db2 change log inventory (DSNJJU003) to update DDF-related information in the BSDS. For more information, see Job DSNFIJUL: update DDF related BSDS information.

Subsystem parameters (Introduction to Db2 for z/OS)
Directory of subsystem parameters, panel fields, and application default values
Bootstrap data set (Introduction to Db2 for z/OS)

Recording Db2 data to SMF (optional)
When you enable Db2 data sharing, you can specify whether Db2 statistical, accounting, and audit trace data is to be collected.

Procedure
To specify the data that is to be collected:

1. Choose from the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To have Db2 collect statistical information</td>
<td>Accept the default (YES class 1) for the SMF STATISTICS option on installation panel DSNTIPN. To collect statistical information for deadlock or timeout, specify class 3. To collect information about DDF error conditions, specify class 4.</td>
</tr>
<tr>
<td>To have Db2 collect accounting information</td>
<td>Accept the default (1) or specify * (all classes) for the SMF ACCOUNTING option on installation panel DSNTIPN.</td>
</tr>
<tr>
<td>To have Db2 collect auditing information</td>
<td>Specify * (all classes) for the AUDIT TRACE option on installation panel DSNTIPN.</td>
</tr>
</tbody>
</table>

In all cases, Db2 invokes a trace, passing the data it collects to the System Management Facility (SMF) of z/OS.
Db2 also passes performance data to SMF whenever an accounting, statistics, or audit trace is successfully started or stopped. Db2 can also record other performance data. After you complete the installation process, you can use commands to have Db2 record performance data for subsystem events.

2. If, during installation, you requested that Db2 pass accounting and statistics data to SMF, update the SMFPRMxx member of SYS1.PARMLIB as follows:
   - Specify the ACTIVE parameter.
   - Specify STC,NODETAIL for the SUBSYS parameter.
   - Specify the proper TYPE subparameter of the SYS and SUBSYS parameters.

What to do next
During Db2 execution, you can use the SMF SET or SS command to alter the SMF parameters. For example, you can record the statistics trace class 1 IFCIDs 0001, 0002, and 0202 (SMF record type 100); accounting trace class 1 IFCIDs 0003 and
0239 (SMF record type 101); and all other Db2 trace records (SMF record type 102) to SMF. To record this information, issue the following command:

SYS(TYPE(100:102))

For Db2 to pass data to SMF, you must allocate an adequate supply of SMF buffers. The default buffer settings are probably insufficient.

You can specify SMF buffering on the VSAM BUFSP parameter of the Access Method Services DEFINE CLUSTER statement. Do not use the default settings if Db2 data is sent to SMF. Specify CISZ(4096) and BUFSP(81920) on the DEFINE CLUSTER statement for each SMF VSAM data set. These values for CISZ and BUFSP are the minimum requirement for Db2. You might need higher values for CISZ and BUFSP, depending on the requirements of all your z/OS subsystems.

You can also code an IEFU84 SMF exit routine to process the records that are produced.

Related tasks:

- Controlling traces (Db2 Administration Guide)

Related reference:

- START TRACE (Db2) (Db2 Commands)
- z/OS MVS Initialization and Tuning Guide

Related information:

- z/OS MVS Initialization and Tuning Reference

### Connecting Db2 to TSO

If you are enabling data sharing, connect Db2 to TSO after you define the initialization parameters. This step is necessary only if you specified a group attachment name.

**About this task**

Access to TSO is required to support the interactive component of Db2 (DB2I). Access to TSO also allows batch applications to access Db2 when those batch programs are executed under the TSO terminal monitor program (TMP).

**Procedure**

To attach Db2 to TSO:

1. Make Db2 load modules available to TSO and batch users.
2. Make Db2 CLISTs available to TSO and batch users.
3. Make PL/I options available (if applicable).
4. Make panels, messages, and load modules available to ISPF and TSO.
5. Connect the DB2I panels to the ISPF Main Panel.

**Related tasks:**

- Make Db2 load modules available to TSO and batch users
- Make Db2 CLISTs available to TSO and batch users (DSNTIJVC)
- Ensure that PL/I options are available
- Make DB2I panels, messages, and load modules available to ISPF and TSO
- Connect DB2I panels to the ISPF main panel
Defining Db2 to z/OS for enabling data sharing

After you run job DSNTIJMV, you might need to define Db2 in SYS1.PARMLIB. This step is only necessary if you are changing the command prefix, or adding or changing the group attachment name.

Before you begin

Run job DSNTIJMV, as described in Identifying Db2 to z/OS: DSNTIJMV.

About this task

Job DSNTIJMV creates new subsystem definitions in the IEFSSNxx member.

Procedure

To define Db2 to z/OS:

Do an initial program load (IPL) of z/OS. You must do this action to apply any Db2 or IRLM subsystem parameter changes in the IEFSSNxx member. After the IPL for z/OS completes, message DSN3100I is displayed on the z/OS console, stating that Db2 is ready for the START command. Alternatively, to apply any Db2 or IRLM subsystem parameter changes until the next IPL, you can use SETSSI commands to delete the subsystem, and add a subsystem with the modified parameters.

For example, the first command below deletes subsystem DSN1. The second command adds a new subsystem named DSN1 that has these characteristics:

- Command prefix -DSN
- Command prefix scope S (started)
- Group attachment name DB0A

SETSSI DELETE,SUB=DSN1,FORCE
SETSSI ADD,SUB=DSN1,INITRTN=DSN3INI,INITPARM="DSN3EPX,-DSN,S,DB0A"

What to do next

Start Db2, as described in Starting the Db2 subsystem.

Related reference:

- START DB2 (Db2) (Db2 Commands)

Related information:

- DSN3100I (Db2 Messages)

Starting the Db2 subsystem

During the process of enabling data sharing, start the Db2 subsystem after you define Db2 to z/OS.

Procedure

To start Db2:

1. Start the IRLM, if you have not requested that Db2 automatically start the IRLM. You must start IRLM before you start Db2.

   Use the following command:

   START irlmproc
where *irlmproc* is the name that you specified for the PROC NAME option on IRLM Panel 1 (DSNTIPI).

If you specified YES for the AUTO START option on IRLM Panel 1 (DSNTIPI), Db2 starts the IRLM automatically.

2. Start Db2 from the z/OS console. The simplest form of the following command is:

```
-DSN1 START DB2
```

where *-DSN1* is the subsystem command prefix that you defined for Db2.

When you issue the START DB2 command with this command format, Db2 uses the subsystem parameter module that is specified in the *ssnmMSTR* start-up JCL procedure in SYS1.PROCLIB.

If you need to change the name of the subsystem parameter module, you can edit the *ssnmMSTR* JCL member in SYS1.PROCLIB. Alternatively, you can override the default subsystem parameter module name by using the PARM option, as follows:

```
-DSN1 START DB2,PARM(DSNZPxxx)
```

If you have not run job DSNTIJRT, and the administrative task scheduler starts, the administrative task scheduler issues error message DSNA679I to report SQLCODE -805 until you run job DSNTIJRT. The administrative task scheduler starts if you specified an address space procedure name in the ADMIN SCHEDULER field on installation panel DSNTIPX.

If Db2 starts successfully, two to five address spaces also start. These address spaces are *ssnmMSTR* and *ssnmDBM1*, possibly *ssnmDIST*, possibly *ssnmADMT*, and *irlmproc*, where *ssnm* is the Db2 subsystem name, *ssnmADMT* is the Db2 administrative task scheduler procedure name, and *irlmproc* is the IRLM procedure name.

If Db2 starts successfully, the series of RESTART messages that you receive concludes with these two messages:

```
DSNR002I   RESTART COMPLETED
DSN9022I   DSNYASCP ' -DSN1 START DB2' NORMAL COMPLETION
```

When you start Db2 12 for the first time, Db2 issues message DSNT501I with reason code 00C900A6. This message is expected. When you run job DSNTIJTC, which is required to complete the tailoring of the Db2 catalog, the cause of this message is corrected. Job DSNTIJTC is described in a later installation step.

After you start Db2, identify unusual conditions for databases with the command:

```
-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT
```

If Db2 does not start successfully, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. Ensure that the DSNTIJID, DSNTIJIE, DSNTIJIF, DSNTIJIN, and DSNTIJUZ jobs ran correctly. Also, verify that the subsystem parameter member that you specified (or allowed by default) when you started Db2 is the one that is built by job DSNTIJUZ. Check the JCL for the Db2 startup procedure.

If you use distributed data facility, you must define VTAM before DDF can start. However, you do not need to have TCP/IP configured to start DDF. In addition, transactions such as those from DDF or CICS may fail because work files have not yet been defined.

3. Optional: Start TSO. After Db2 starts, it is available to TSO. If you want to use the TSO SUBMIT command to do housekeeping and installation verification, you must start TSO (if it is not already started).
Creating the default storage group and defining temporary work files: DSNTIJTM

After you start Db2 as part of the process for enabling data sharing, you must create the default storage group for user-defined Db2 tables that are not specifically assigned to a storage group. You must also define the database for temporary work files.

Procedure

To create the default storage group and define temporary work files:

Run job DSNTIJTM.

If job DSNTIJTM runs successfully, it produces the return codes that are shown in the following table:

Table 68. DSNTIJTM return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIAS</td>
<td>IKJEFT01</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTICR</td>
<td>IKJEFT01</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIST</td>
<td>IKJEFT01</td>
<td>0000</td>
</tr>
</tbody>
</table>

Results

The SYSDEFLT storage group is the default storage group, and it is used for your database, table space, and table definitions that are not related to a specific storage group. By default, it is also used for the work file database and for work file table spaces that are created during Db2 installation and migration. When DSNTIJTM creates the SYSDEFLT storage group, it completes the following grants:

- Grants use of the default buffer pool and storage group to PUBLIC or the authorization IDs that you specified in the INSTALL GRANTEE(S) field of panel DSNTIPG.
- Grants use of the SYSDEFLT storage group to PUBLIC or to the authorization IDs that you specified in the INSTALL GRANTEE(S) field of panel DSNTIPG. This access is necessary to give users the ability to implicitly create table spaces in the default database.
- Grants authority to create tables and table spaces in the default database to PUBLIC or to the authorization IDs that you specified in the INSTALL GRANTEE(S) field of panel DSNTIPG.

What to do next

For data sharing installations, the work file database name is the name that was specified in the WORK FILE DB field on installation panel DSNTIPK. After creating the work file database, the DSNTIJTM job creates the work file table spaces as specified on installation panel DSNTIP9.

You can increase the number of additional temporary work file table spaces by increasing the values in the work file table space fields on panel DSNTIP9, particularly if you expect a large amount of sorting at your site. Additional temporary work file table spaces can improve Db2 performance by reducing device contention among applications. These additional work files also can be used for sorting indexes on large tables during index creation.
You can choose to have job DSNTIJTM create these additional table spaces, or you can create them after you run DSNTIJTM. To create additional work file table spaces after you run job DSNTIJTM, use the DSNTWFG exec in job step DSNTIST. See the comment block for job step DSNTIST for information about the parameters for DSNTWFG.

If you do not run the installation CLIST, customize DSNTIJTM or the DSNTWFG exec to create at least one 4 KB and one 32 KB work file table space.

Creating an image copy of the Db2 directory and catalog: DSNTIJJC (optional)

If you decide to image copy the Db2 directory and catalog, use the DSNTIJJC job that was generated during the installation or migration of the originating member. You do not need to make an image copy of the Db2 catalog or user data sets for recovery because Db2 uses image copies that were made before you enabled data sharing.

Procedure

To create an image copy of the directory and catalog:

1. Examine job DSNTIJJC for the following information:
   - The device type or unit name.
     The default value is SYSDA. If this is incorrect for your site, correct it.
   - Expiration date or retention period. You can add a retention period or an expiration date to the job.
   - The user on the JOB statement. Ensure that the user is authorized. If Db2 is started under ACCESS(MAINT), this must be the same user that you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP1.

2. Run job DSNTIJJC.

   The DSNTIJJC job contains a list of all the Db2 directory and catalog table spaces. When you run job DSNTIJJC, it invokes the Db2 image copy utility to copy these table spaces to disk or tape, depending on the value that you select for [IC DATA SET UNIT NAME field]. Having copies of table spaces enables you to recover the Db2 catalog and Db2 directory in case of a failure.

3. If the DSNTIJJC job fails or abends, verify that no problems exist with the tape or disk setup for image copy. If you find no problems with the setup, examine the utility job output (JOBLOG) or the console log for problems. For example, look for I/O errors or incorrect sizes.

What to do next

Run the DSNTIJJC job periodically, perhaps daily or weekly, to reduce the amount of time required for recovering the directory or catalog. The copied data and log data sets are needed for recovery.

During the ISPF tailoring session, you named one or two IDs to have installation SYSADM authority. One of these users can now grant various levels of authority to other users. You can use SPUFI or a job similar to DSNTIJSG to perform the authorization. To grant SYSOPR authority to all primary and secondary user IDs that issue Db2 commands from the z/OS console or TSO SDSF, issue the following statement in SPUFI:

```
GRANT SYSOPR TO userid
```
Verifying that enabling data sharing was successful (optional)

If you ran the complete set of installation verification procedure (IVP) sample jobs after you migrated to Db2 12, you should not need to run these jobs again.

About this task

When you start the originating member, Db2 checks your coupling facility, group, and member definitions and verifies that data sharing is enabled. You can also verify that the group has been correctly established by issuing the Db2 command DISPLAY GROUP after the originating member has completed startup.

Procedure

To verify that enabling data sharing was successful:

Run the verification jobs that are described in Chapter 10, “Verifying successful Db2 12 installation or migration with sample applications,” on page 733.

Adding a new data sharing member

After you install a new data sharing group or enable an existing Db2 subsystem for data sharing, you can add new data sharing members. Always add a member as a new installation. After being added to the group, the new member uses the Db2 catalog of the originating member.

About this task

When adding a new member to a group, you might need to make changes during installation to allow more XCF groups, or more XCF members per group. Or, you might need to “widen” the locks in the IRLM lock structure (for example, if the structure was initially allocated with a maximum of seven members, and the eighth member is joining the group).

Db2 does not have an automatic way to “merge” catalogs and resolve naming conflicts. If you have applications that currently run on several existing Db2 subsystems, your migration plan might include procedures for moving the relevant data and applications from those Db2 subsystems onto one or more of the group members. Your migration plan should also include procedures for resolving any catalog naming conflicts that result.

Related concepts:
Merging existing Db2 data into the group

Related tasks:
Installing Db2

Related information:
IEFSSNxx (subsystem definitions) - keyword parameter form (MVS Initialization and Tuning Reference)

Running the CLIST for adding a new member

The process of adding a new data sharing member involves running some of the installation jobs. Before you run the installation jobs, tailor them to your environment by running the installation CLIST.
About this task

Jobs that the add-member CLIST tailors:

- DSNTIJIN
- DSNTIJMV
- DSNTIJID
- DSNTIJDE
- DSNTIJTM
- DSNTIJJUZ
- DSNTIJFT
- DSNTIJGF

Procedure

To tailor the installation jobs for adding a new member:

1. Invoke the installation CLIST as described in Tailoring Db2 12 installation and migration jobs with the CLIST.
2. On installation panel DSNTIPA1, specify the following values:
   - INSTALL TYPE ==> INSTALL
   - DATA SHARING ==> YES
   - INPUT MEMBER NAME ==> originating member's output PDS member
   - OUTPUT MEMBER NAME ==> new member's output PDS member
3. On installation panel DSNTIP0A, specify option 2 for adding a member.
4. On installation panel DSNTIPK, specify a new member name:
   - MEMBER NAME ==> new member name
   - Verify that the new member name is unique within the z/OS Parallel Sysplex.
   - Installation job DSNTIJMV edits the ssnnMSTR startup procedure with the member name that you specify on panel DSNTIPK.
5. Complete the installation panels, specifying parameters according to the Db2 subsystem parameters guidelines.

Recommendation: Rename the tailored SDSNSAMP data set for each member. This data set contains tailored JCL for each member, including jobs that are used for disabling and re-enabling data sharing, if that should ever become necessary. If you do not rename the SDSNSAMP data set, it is overwritten when you install a new member. Rename the data set by choosing a new name for the prefix.NEW.SDSNSAMP data set on installation panel DSNTIPT. For example, use prefix.member_name.SDSNSAMP.

Related concepts:
- Subsystem parameters and application defaults in a data sharing environment
- The Db2 installation CLIST panel session

Define the new Db2 data sharing member to IRLM and MVS:

DSNTIJMV

You must define the new Db2 data sharing member in the system parameter library of the target z/OS system.

Before you begin

The z/OS operating system must be installed at the prerequisite level.

About this task

Job DSNTIJMV demonstrates how to update the SYS1.PARMLIB members to identify Db2 to IRLM and MVS. Although you might edit and run this job, the
more common approach is to examine the changes that DSNTIJMV makes and use
an editor, such as ISPF/PDF, to make the required updates directly to the
SYS1.PARMLIB members.

Because DSNTIJMV cannot anticipate all the necessary updates to identify Db2 to
MVS and IRLM in your z/OS environment, additional updates might be required
for your site. You can complete these updates either by making the updates
directly in SYS1.PARMLIB members, or by editing the DSNTIJMV job.

Do not overwrite the entries for Db2 and IRLM in the program properties table
(PPT). Note that PPT entries should no longer be defined in
SYS1.PARMLIB(SCHEDxx). Default entries are supplied in the
SYS1.LINKLIB(IEFSDPPT) load module, which is provided by z/OS. For details of
the PPT entries, see Program properties table (PPT) (MVS Initialization and Tuning
Reference)

Procedure

To identify the new Db2 data sharing member to IRLM and MVS:
1. Examine the DSNTIJMV job and study the updates that it makes.
2. Update the IEFSSNxx member of SYS1.PARMLIB by editing it directly, or by
tailoring and running job DSNTIJMV. The IEFSSNxx member contains an entry
for every z/OS subsystem. Db2 adds one entry for Db2 and two entries for
IRLM in the list. The second IRLM entry, whose subsystem name is JRLM,
simplifies adding a second IRLM to your system if the first is damaged. You
must provide your own procedure to add JRLM. Unique names must be used
for each entry.

z/OS provides subsystem entries for Db2 and IRLM in IEFSSN00. Examine
these entries to determine whether they are appropriate for your needs. Make
sure that a subsystem name appears only once in the subsystem name list.

Place the SMS line as the first line in an IEFSSNxx member. Place the primary
system’s record (JES2 or JES3) record next. Other products might affect the
position of the entries during system initialization. The Db2 line should come
after SMS, the JES subsystem, and other vendor products.

The Db2 entry has the following format:

```
SUBSYS SUBNAME(ssname) INITRTN(DSN3INI)
   INITPARM(’DSN3EPX,cmd-prefix<,scope<,group-attach<,subgroup-attach>>>’)

ssname
   The Db2 subsystem name.

DSN3INI
   The name of the Db2 load module that z/OS invokes during master
   scheduler initialization. This module must be located in a link list data set
   (or in SYS1.LINKLIB).

DSN3EPX
   The name of the Db2 load module that responds to Db2 requests that are
   received from the z/OS subsystem interface. (Db2 can be active or inactive
   when the requests are received.) This module must be located in a link list
   data set (or in SYS1.LINKLIB).

cmd-prefix
   The one- to eight-character command prefix.

   The first character of the command prefix must be one of the following
   characters: @ # . / ’ ) + - . ą < | & ! ; % _ ? : ”. The remaining characters
of the command prefix must be one of the above characters, A-Z, or 0-9. Do not use the JES2 backspace character or command prefix character. The default is the hyphen (-).

Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or z/OS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix beginning from the first character. For example, you cannot assign '-' to one subsystem and '-Db2A' to another. Similarly, you also cannot assign '?' to one subsystem and '?Db2A' to another. You can assign '-Db2A' and '-Db2B' to different Db2 subsystems.

**scope**

The one-character scope for the command prefix. Db2 registers its command prefix with the operating system. When this is done, the scope of the command prefix is controlled by the value you choose:

- **S**: Started; and the prefix is registered with Sysplex scope at Db2 startup instead of during z/OS IPL. This is the default.

  **Recommendation**: Choose S, which allows you to have a single IEFSSNxz PARMLIB member that all z/OS systems in the Sysplex can use. Choosing S also simplifies the task of moving a Db2 from one system to another; you can stop Db2 on one z/OS and start it up on another without having to re-IPL the system.

- **M**: z/OS system scope; the prefix is registered during z/OS IPL.

- **X**: Sysplex scope; the prefix is registered during z/OS IPL. As a result, this Db2 cannot be restarted on another z/OS without changing the definitions and re-IPLing both z/OS systems.

**group-attach**

The group attachment name, which is used for data sharing. You can specify this on installation panel DSNTIPK.

**subgroup-attach**

is the name of the subgroup attachment, which is used in data sharing environments to provide more control over how TSO, CAF, RRSAF, JDBC, ODBC, and Db2 utilities connections find active Db2 subsystems. You can specify this value by entering a subgroup name in the SUBGRP ATTACH field on installation panel DSNTIPK.

---

**Define address space startup procedures for the new Db2 data sharing member: DSNTIJMA**

You must define the address space startup procedures for the new Db2 data sharing member.

**Before you begin**

The authorization ID that runs the job must have authority to update the target system PROCLIB. You can enter that library name on installation panel DSNTIPW in the SYSTEM PROCEDURES field. You might want to use a procedure library other than SYS1.PROCLIB for the procedures.

Examine the updates from job DSNTIJMA carefully before you run it.
About this task

Because different sites have different requirements, the jobs in the following steps cannot anticipate all the necessary updates to the target system procedure library in your z/OS environment. You can complete these updates either by making the updates directly in the target system procedure library, or by editing and running job DSNTIJMA.

Procedure

To define the address space startup procedures for the new Db2 data sharing member in the system procedure libraries:

Tailor and run job DSNTIJMA. Job DSNTIJMA defines the startup procedures in the target system procedure library for the following Db2 address spaces, where ssnn is the subsystem name:

- ssnnMSTR for Db2 system services
- ssnnDBM1 for Db2 database services
- ssnnDIST for the Db2 distributed data facility (DDF)
- ssnnADMT for the Db2 administrative task scheduler
- IRLMPROC, or a user-defined procedure for IRLM

Examine the size of the private area on the Db2 startup procedures. If necessary, modify the procedures to satisfy the requirements for environmental descriptor manager (EDM) pool size, buffers, number of data sets open, and amount of available private address space. For more information, see Calculating working storage requirements.

The STEPLIB concatenation of the ssnnDBM1 address space procedure includes a commented-out DD for the IBM Language Environment data sets SCEERUN and SCEERUN2. If your system does not include SCEERUN and SCEERUN2 in the system link list, you must uncomment this DD.

Defining system data sets: DSNTIJIN

When job DSNTIJIN is tailored for adding a new member, it defines the new BSDS and active log data sets for that member to use SHAREOPTIONS (2 3). It does not define catalog and directory data sets.

Procedure

To define system data sets:

1. Check the DEFINE CLUSTER statements in job DSNTIJIN to ensure that they allocate adequate disk space for your system.

   **Recommendation:** For recovery purposes, place system data sets such as the Db2 recovery log and the VSAM catalog on different disk volumes. Because these data sets are used frequently, do not migrate them by using DFSMSShsm.

2. Run job DSNTIJIN.

   If DSNTIJIN runs successfully, it produces return codes of 0 for all DEFINE statements and steps. Check any VSAM messages carefully.

3. If job DSNTIJIN fails or abends, run the DSNTIJDE job for the member to delete its BSDS and active log data sets. Then rerun the DSNTIJIN job for the member to re-create those data sets.

Related concepts:
Allocating and extending data sets (Db2 Performance)

### Defining Db2 initialization parameters for new data sharing members

For new data sharing members, you must define Db2 12 initialization parameters in the subsystem parameter module (DSNZP:xxx) and application defaults load module (dsnhdcecp).

#### About this task

The subsystem parameter module (DSNZP:xxx) and application defaults module (dsnhdcecp) names are specified in the following DSNITPO3 installation panel fields:
- **PARAMETER MODULE field**
- **APPL DEFAULTS field**

#### Procedure

To define Db2 initialization parameters for new data sharing members, complete the following steps:

1. Run job DSNTIJUZ. Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZP:xxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSYP.

   You might need to make the following adjustments before running the job:
   - If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix:SDSNEXIT and prefix:SDSNLOAD, you can move the SYSLMOD output to that library.
   - If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
   - If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This action ensures that members DSNFSYSP, DSNJARVP, DSNJLOGP, DSNTSPRM, DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:
     - SET BOUNDARY (GLOBAL).
     - LIST DDDEF (SMPTLIB).

     Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

   When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions.

   For more information, see [Job DSNTIJUZ: define the Db2 data-only subsystem parameter module](#)

2. Run job DSNTIJUL. Job DSNTIJUL runs the Db2 change log inventory (DSNJU003) to update DDF-related information in the BSDS. For more information, see [Job DSNTIJUL: update DDF related BSDS information](#)

---

Subsystem parameters (Introduction to Db2 for z/OS)

Application programming default values (Introduction to Db2 for z/OS)
Initializing system data sets: DSNTIJID
If you are adding a new data sharing member, you must initialize its BSDS and active log data sets after you define them and the member’s Db2 initialization parameters.

Procedure
To initialize system data sets:

Run job DSNTIJID.
- If the job runs successfully, it produces a return code of 0. Otherwise, check for and resolve error conditions that might be reported by VSAM messages.
- If the job fails or abends, rerun jobs DSNTIJDE, DSNTIJIN, DSNTIJUZ, and DSNTIJID for that member.

Results
Job DSNTIJID records the active log data set names to the BSDS, formats the active log data sets, and initializes the Db2 directory table spaces and indexes.

Defining user authorization exit routines: DSNTIJEX (optional)
When you add a new data sharing member, you should not have to make changes to the Db2 exit routines. However, if the new member is using its own separate load module or exit library, you must define the exit routines in the new library.

Procedure
To define user authorization exit routines:
1. Modify job DSNTIJEX to reference the correct library.
2. Run job DSNTIJEX.
   - If job DSNTIJEX runs successfully, it produces return codes of 4.

Related concepts:
- Connection routines and sign-on routines (Managing Security)

Recording Db2 data to SMF (optional)
When you add a Db2 data sharing member, you can specify whether Db2 statistical, accounting, and audit trace data is to be collected.

Procedure
To specify the data that is to be collected:
1. Choose from the following options:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To have Db2 collect statistical information</td>
<td>Accept the default (YES class 1) for the SMF STATISTICS option on installation panel DSNTIPN. To collect statistical information for deadlock or timeout, specify class 3. To collect information about DDF error conditions, specify class 4.</td>
</tr>
<tr>
<td>To have Db2 collect accounting information</td>
<td>Accept the default (1) or specify * (all classes) for the SMF ACCOUNTING option on installation panel DSNTIPN.</td>
</tr>
<tr>
<td>To have Db2 collect auditing information</td>
<td>Specify * (all classes) for the AUDIT TRACE option on installation panel DSNTIPN.</td>
</tr>
</tbody>
</table>

In all cases, Db2 invokes a trace, passing the data it collects to the System Management Facility (SMF) of z/OS.

Db2 also passes performance data to SMF whenever an accounting, statistics, or audit trace is successfully started or stopped. Db2 can also record other performance data. After you complete the installation process, you can use commands to have Db2 record performance data for subsystem events.

2. If, during installation, you requested that Db2 pass accounting and statistics data to SMF, update the SMFPRMxx member of SYS1.PARMLIB as follows:
   - Specify the ACTIVE parameter.
   - Specify STC,NODETAIL for the SUBSYS parameter.
   - Specify the proper TYPE subparameter of the SYS and SUBSYS parameters.

What to do next

During Db2 execution, you can use the SMF SET or SS command to alter the SMF parameters. For example, you can record the statistics trace class 1 IFCIDs 0001, 0002, and 0202 (SMF record type 100); accounting trace class 1 IFCIDs 0003 and 0239 (SMF record type 101); and all other Db2 trace records (SMF record type 102) to SMF. To record this information, issue the following command:

```
SYS(TYPE(100:102))
```

For Db2 to pass data to SMF, you must allocate an adequate supply of SMF buffers. The default buffer settings are probably insufficient.

You can specify SMF buffering on the VSAM BUFSP parameter of the Access Method Services DEFINE CLUSTER statement. Do not use the default settings if Db2 data is sent to SMF. Specify CISZ(4096) and BUFSP(81920) on the DEFINE CLUSTER statement for each SMF VSAM data set. These values for CISZ and BUFSP are the minimum requirement for Db2. You might need higher values for CISZ and BUFSP, depending on the requirements of all your z/OS subsystems.

You can also code an IEFU84 SMF exit routine to process the records that are produced.

Related tasks:

- Controlling traces (Db2 Administration Guide)

Related reference:

- START TRACE (Db2) (Db2 Commands)
- z/OS MVS Initialization and Tuning Guide
Establishing subsystem security: DSNTIJRA (optional)

Db2 can control access to data within a Db2 subsystem. Db2 also works with outside security systems, such as RACF, that control access to the Db2 subsystem.

About this task

Most of the tasks to establish subsystem security for a data sharing group need to be completed only after the originating member is installed. For other members, you must set up only the security of the administrative task scheduler of those members.

Procedure

To set up subsystem security for data sharing members:

Customize and run job DSNTIJRA to complete the following tasks for each member of the group:

- Define the administrative task scheduler started task module to RACF program control
- Define the administrative task scheduler as a trusted context in RACF

Connecting Db2 to IMS (optional)

Connecting Db2 to IMS is an optional part of adding a new data sharing member. Connecting Db2 to IMS requires coordination with your company's group that supports IMS systems.

Procedure

To connect the IMS attachment facility:

1. Make Db2 load modules available to IMS
2. Define Db2 to IMS
3. Define new programs and transactions to IMS
4. Prepare IMS applications for Db2
5. Depending on your site, you might also need to:
   - Define Db2 plans for IMS applications
   - Generate a user language interface

Connecting CICS to Db2 (optional)

When you add a new member, you have the option of connecting CICS to the member.
About this task

To connect Db2 to CICS, you must regenerate several CICS tables with additional entries. Coordinate this connection with your CICS support group.

Related tasks:

- Defining the CICS Db2 connection

Defining Db2 to z/OS for adding data sharing members

After you run job DSNTIJMV, you must define Db2 in SYS1.PARMLIB.

Before you begin

Enable data sharing on the originating member, as described in Enabling Db2 data sharing.

About this task

Job DSNIJMV creates new subsystem definitions in the IEFSSN.xx member of the SYS1.PARMLIB library.

Procedure

To define Db2 to z/OS:

Complete one of the following actions:

- IPL z/OS.
  
  During the z/OS IPL, message DSN3100I appears on the z/OS console, stating that Db2 is ready for the START DB2 command.

- Use the z/OS command SETSSI to dynamically add the new Db2 and IRLM subsystems. For example, the following two commands can be used to add subsystems for Db2 and IRLM to MVS3:

  - `RO MV3,SETSSI ADD,SUB=DB3A,INITRTN=DSN3INI,INITPARM='DSN3EPX,-DB3A,S,SB0A'
  - `RO MV3,SETSSI ADD,SUB=DJ3A`

  Recommendation: Add z/OS to the IEFSSN.xx member so that it can be used on a subsequent IPL.

Related tasks:

- Strategy for creating a Db2 data sharing group

Related reference:

- z/OS SETSSI Command (MVS System Commands)

Related information:

- DSN3100I (Db2 Messages)

Starting the Db2 subsystem

During the process of adding a data sharing member, start the Db2 subsystem after you define Db2 to z/OS.

Procedure

To start the Db2 member:
1. Start the IRLM, if you have not requested that Db2 automatically start the IRLM. You must start IRLM before you start Db2.

Use the following command:

```
START irlmproc
```

where `irlmproc` is the name that you specified for the PROC NAME option on IRLM Panel 1 (DSNTIPI).

If you specified YES for the AUTO START option on IRLM Panel 1 (DSNTIPI), Db2 starts the IRLM automatically.

2. Start Db2 from the z/OS console. Use the following command:

```
-DSN1 START DB2
```

where `(-DSN1)` is the subsystem command prefix that you defined for Db2.

Db2 uses the subsystem parameter module that is specified in the start-up JCL procedure in SYS1.PROCLIB:

```
//IEFPROC EXEC PGM=DSNYASCP,PARM='ZPARM(DSNZPxxx)', ...
```

where `DSNZPxxx` is the value that you specified for the PARAMETER MODULE field on installation panel DSNTIPO3.

If you need to change the name of the `DSNZPxxx` module, you can edit SYS1.PROCLIB. Alternatively, you can override the `DSNZPxxx` name by using the PARM option as follows:

```
-DSN1 START DB2,PARM(DSNZPxxx)
```

If you have not run yet job DSNTIJRT on any member of the group, and the administrative task scheduler for this member starts, the administrative task scheduler issues error message DSNA679I to report SQLCODE -805 until you run job DSNTIJRT. The administrative task scheduler starts if you specified an address space procedure name in the ADMIN SCHEDULER field on installation panel DSNTIPX.

If Db2 starts successfully, two to five address spaces also start. These address spaces are `ssnmMSTR` and `ssnmDBM1`, possibly `ssnmDIST`, possibly `ssnmADMT`, and `irlmproc`, where `ssnm` is the Db2 subsystem name, `ssnmADMT` is the Db2 administrative task scheduler procedure name, and `irlmproc` is the IRLM procedure name.

If Db2 starts successfully, the series of RESTART messages that you receive concludes with these two messages:

```
DSNR002I  RESTART COMPLETED
DSN9022I  DSNYASCP '-DSN1 START DB2' NORMAL COMPLETION
```

After you start Db2, identify unusual conditions for databases with the command:

```
-DSN1 DISPLAY DATABASE(*) SPACENAM(*) RESTRICT
```

If Db2 does not start successfully, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. Ensure that the DSNTIJID, DSNTIJIN, and DSNTIJUZ jobs ran correctly. Also, verify that the subsystem parameter member that you specified (or allowed by default) when you started Db2 is the one that is built by job DSNTIJUZ. Check the JCL for the Db2 startup procedure.

If you use distributed data facility, you must define VTAM before DDF can start. However, you do not need to have TCP/IP configured to start DDF. In addition, transactions such as those from DDF or CICS may fail because work files have not yet been defined.
3. Optional: Start TSO. After Db2 starts, it is available to TSO. If you want to use the TSO SUBMIT command to do housekeeping and installation verification, you must start TSO (if it is not already started).

Results

Transactions that run on a new member might initially fail because work files are not defined until you run that member’s job DSNTIJTM.

Creating the default storage group, defining temporary work files, and binding Db2 REXX Language Support: DSNTIJTM

After you start Db2 when you add a data sharing member, you must activate its buffer pools and create its work file database.

Procedure

To define the new member’s buffer pools and create its work file database:

Run job DSNTIJTM.
If job DSNTIJTM runs successfully, it produces the return codes that are shown in the following table:

Table 69. DSNTIJTM return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIAB</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIAS</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTICR</td>
<td>(none)</td>
<td>0000</td>
</tr>
<tr>
<td>DSNTIST</td>
<td>(none)</td>
<td>0000</td>
</tr>
</tbody>
</table>

Results

DSNTIJTM defines the new member’s initial buffer pools according to the sizes that are specified on installation panels DSNTIP1 and DSNTIP2. It also creates the work file database for the new member according to the values that you specified on panel DSNTIP9.

DSNTIJTM also provides some cleanup to ease reruns.

What to do next

For data sharing installations, the work file database name is the name that was specified in the WORK FILE DB field on installation panel DSNTIPK. After creating the work file database, the DSNTIJTM job creates the work file table spaces as specified on installation panel DSNTIP9.

You can increase the number of additional temporary work file table spaces by increasing the values in the work file table space fields on panel DSNTIP9, particularly if you expect a large amount of sorting at your site. Additional temporary work file table spaces can improve Db2 performance by reducing device contention among applications. These additional work files also can be used for sorting indexes on large tables during index creation.
You can choose to have job DSNTIJTM create these additional table spaces, or you can create them after you run DSNTIJTM. To create additional work file table spaces after you run job DSNTIJTM, use the DSNTWFG exec in job step DSNTIST. See the comment block for job step DSNTIST for information about the parameters for DSNTWFG.

If you do not run the installation CLIST, customize DSNTIJTM or the DSNTWFG exec to create at least one 4 KB and one 32 KB work file table space.

Related concepts:
- Work file database storage requirements

Creating an image copy of the Db2 directory and catalog: DSNTIJIC

After you add a data sharing member, you need to create a backup copy of the Db2 directory and catalog. Use the DSNTIJIC job that was generated during the installation or migration of the originating member to create the image copy.

Procedure

To create an image copy of the directory and catalog:
1. Run job DSNTIJIC.
   - The DSNTIJIC job contains a list of all the Db2 directory and catalog table spaces. When you run job DSNTIJIC, it invokes the Db2 image copy utility to copy these table spaces to disk or tape, depending on the value that you select for the [IC DATA SET UNIT NAME field]. Having copies of table spaces enables you to recover the Db2 catalog and Db2 directory in case of a failure.
2. If the DSNTIJIC job fails or abends, verify that no problems exist with the tape or disk setup for image copy. If you find no problems with the setup, examine the utility job output (JOBLOG) or the console log for problems. For example, look for I/O errors or incorrect sizes.

What to do next

Run the DSNTIJIC job periodically, perhaps daily or weekly, to reduce the amount of time required for recovering the directory or catalog. The copied data and log data sets are needed for recovery.

Related concepts:
- Privileges and authorities (Managing Security)

Verifying that adding a data sharing member was successful

When you complete the procedure to add a new data sharing member, you should verify that the member was added successfully.

Procedure

To verify that adding a data sharing member was successful:

Run the verification jobs that are described in Chapter 10, “Verifying successful Db2 12 installation or migration with sample applications,” on page 733.
What to do next

After you successfully add all the data sharing members for the group, you should **Test the data sharing group**.

Ensuring that the WLM address space is defined and available

After you verify that the data sharing member was added successfully, you must ensure that the WLM address space for the member is defined and available.

About this task

When the WLM address space is defined correctly and available, the WLM address spaces can start without errors when connections land on any member of the group.

Procedure

To verify that the WLM address space is defined and available:

Update the JCL startup procedure for the WLM application environment by using one of the following methods:

- Add the WLM application environments that are specific for each member of the group.
- Modify existing definitions to work for any member of the group.

Related concepts:

**Special considerations for WLM-managed stored procedures in coexistence**

Merging existing Db2 data into the group

Although Db2 cannot automatically merge catalogs, you can merge existing Db2 subsystems into a data sharing group. Consider carefully a decision to merge existing Db2 subsystems.

Merging is a very complicated process. It involves not only the physical issue of moving data, but also many other management issues, including:

- Naming conventions for users, plans, packages, databases, tables, views, and more
- Authorization techniques
- Backup and recovery conventions
- Availability practices

Before you consider merging existing Db2 subsystems into a single data sharing group, ask yourself the following question: Why are the Db2 subsystems separate now?

Merging subsystems

You can merge existing Db2 subsystems into a single Db2 data sharing group.

Procedure

To merge existing subsystems into a single Db2 data sharing group, complete the following steps:

1. Choose one of the subsystems to be the originating member.
2. Move data and data definitions from the other Db2 subsystems to the subsystem that you designated as the originating member. For more information, see [Moving data that you are merging](#).

**Tip:** Use a location alias, remote applications do not have to change aliases during the merging processes. Remote applications use a location name to access the Db2 subsystem. You can define the previous location names of each subsystem as location alias names for the group. After the subsystems are merged, remote applications continue to connect, without changes to the location names in the applications. For more information, see [Specifying a location alias name for multiple sites (Db2 Application Programming and SQL)](#).

3. Enable data sharing on the originating member, as described in [Enabling Db2 data sharing](#).

4. Add the other Db2 subsystems as members of the data sharing group, as described in [Adding a new data sharing member](#).

### Merging data

If you have an application that is currently running on independent Db2 subsystems, you might decide that the application will work well in a data sharing group.

In that case, you must move the data and merge the catalog definitions for that data from the independent Db2 subsystems into the data sharing group. Because the Db2 subsystems still exist, you cannot reuse their subsystem names when installing the subsystems as new members into the data sharing group.

Db2 does not provide an automated way to move catalog definitions from an independent Db2 subsystem into the catalog of the data sharing group. If you have procedures and tools to move applications from test to production, or to handle merging databases from enterprise reorganizations or mergers, those same procedures can be used to move applications into the data sharing group.

### Existing distributed applications

If you move existing data to the data sharing group, it is likely that the location name of objects will change.

Existing distributed applications that remotely reference an object by its three-part name must be changed to reference the new name. Aliases that exist on that table must also be dropped and re-created with the new location name.

Applications that contain explicit SQL CONNECT statements that reference an old location name must be modified. Any Db2 plan that uses an old location name for the CURRENTSERVER keyword must be bound again.

### Moving data that you are merging

In the context of Db2 data sharing, a **target** refers to the Db2 subsystem to which you are moving data (the data sharing group). A **source** refers to the Db2 subsystem from which you are moving data.

### Procedure

To move data that you are merging:

1. Decide the method that you will use to move data. Three options are outlined below; you can use a combination of the specified methods. Each of the methods assumes that the objects that are to be created in the target system...
are created with the exact DDL that is used in the source system. Descriptions of methods 1, 2, and 3 are referred to throughout the procedure, and are described in detail below.

- **Method 1: Unload and Reload**: The data that is to be moved is unloaded from the source Db2 subsystem and loaded into the target data sharing group by using the Db2 online utilities, UNLOAD and LOAD. This approach is the most simple because tables are created without concern for object IDs, and data is unloaded or reloaded using conventional techniques. The major disadvantage of this method is that it is the slowest of the three methods, in terms of performance. The unload and reload methods operate against rows of a table, whereas the other techniques operate against pages or data sets as their level of granularity. Additionally, this method requires two to three times the disk space that the other methods require, depending on how you unload or reload (source, target and intermediate unload).

- **Method 2: DSN1COPY with OBID translation**: The data that is to be moved is copied from the source system to the target system using the Db2 offline utility DSN1COPY. DSN1COPY is usually faster than unload and reload because it moves pages of data, rather than rows. This method is more complicated because the object IDs in the source data will need to be translated.

- **Method 3: Use Db2 VSAM data sets from the source system in the target system**: The VSAM data sets that are used to hold the table space and index space data from the source system are used in the target system, without copying any data. Because this method does not require copying the data, it is the fastest of the three methods, in terms of performance. It is, however, the most complicated. Another advantage of this method is that it requires less disk space than the other methods; this method is generally used for very large table or index spaces. Method 3 assumes that all of the disk volumes in the source system used by objects being moved by this method are shared and accessible with the target system.

2. Choose a catalog that belongs to one of the Db2 subsystems to be the “original” catalog for the data sharing group. This Db2 subsystem is considered the originating member of the group.

You should consider many different factors when choosing which Db2 subsystem to be the originating member of the group. For example, it makes sense to choose the member with the most database objects as the originating member in order to minimize the number of objects that you move. However, if all Db2 subsystems are mostly equivalent, and you are not planning to use DSN1COPY, consider the log RBA values of the existing subsystems. Compare the end-of-log RBAs with the high-order 10 bytes (for 10-byte RBAs) or 6 bytes (for 6-byte RBAs) of the time-of-day clock timestamp on each of their systems. This value is called the truncated timestamp. The usual case is that the RBA is less than the truncated timestamp. In this case, any Db2 subsystem is chosen as the originating member.

In the event that the current end-of-log RBA in any of the existing Db2 subsystems is higher than the truncated timestamp value at the time you are ready to enable sharing, you have the following choices:

- Choose the Db2 subsystem that has the highest RBA as the originating member.

- Use DSN1COPY with the RESET option to reset the log RBAs in each data and index page to 0 when you move databases from other Db2 subsystems to the data sharing group.
3. Resolve name conflicts among the objects and authorization IDs in the data sharing group.

4. Create the objects on the target subsystem.

When creating objects while you are using methods 1 or 2 to move the data, the CREATE statements can be entered in any order. Db2 assigns new OBIDs for these objects.

If you are using method 3, you must perform the following steps:

a. Consider creating the data sets with the minimum size settings to save space. The actual target system VSAM data sets created by the DDL execution are not used because the VSAM data sets that are from the source system are used instead. By choosing to create the objects with minimum sizes, you eventually need to issue ALTER TABLESPACE and ALTER INDEX statements to change the PRIQTY and SECQTY after the CREATE. If, however, you are using the same high-level qualifier for the source and target systems (which is not recommended), you must create the objects with DEFINE NO, and take additional steps later. DEFINE NO is needed when keeping the same high-level qualifier, because the VSAM data set name would be identical in the source and target systems. In this case, Db2 would not be able to define the data set on the target system during the create processing because the data set already exists.

b. Query the SYSIBM.SYSTABLES table in the Db2 catalog on the source subsystem to get the table OBID for tables that are within the databases that are being moved. Additionally for both the source and target systems, the DBID and PSID are needed for table spaces (from SYSIBM.SYSTABLESPACE) and the DBID and ISOBID are needed for indexes (from SYSIBM.SYSINDEXES). These will be used to build REPAIR jobs to modify these IDs in the header page, and potentially the first space map page in a later step.

c. Use the OBID clause on the CREATE TABLE statement on the target subsystem to specify an OBID that is the same as the table OBID on the source subsystem. Verify that the OBID that you specify is available. If an OBID is being used for another object within the same database, such as an index or referential constraint, Db2 does not allow you to create the table with the specified OBID.

Recommendation: To help guarantee the availability of OBIDs for all tables within a database, defer the creation of all indexes and referential constraints until all tables are created. All CREATE TABLE statements must have the OBID clause to guarantee that they are assigned the correct OBID. If an explicit table space name is specified, then the CREATE TABLESPACE statement must come immediately before the first CREATE TABLE statement for that table space. This helps prevent a group of CREATE TABLESPACE statements from using up OBIDs that are needed for the tables.

For methods two or three, the objects in the target system will be created with several assumptions that might require the additional steps that are outlined below:

- The VSAM data sets for each new object are created using the ‘I’ prefix for each of the data sets. It is possible that on the source system online REORG has executed, causing the data sets to have a ‘J’ prefix. If this is the case, you will need to code the DSN1COPY for method two, to have the correct ‘J’ data set designation or rename the data set back to the ‘T’ prefix for method three in a later step. You should query the source catalog for column IPREFIX in SYSIBM.SYSTABLEPART for table spaces, and SYSIBM.SYSINDEXPART for indexes, to find any ‘J’ prefix objects.
• If any of the index names involved in the move are greater than eight characters in length, Db2 generates a unique INDEXXSPACE name to be used in naming the VSAM data set. The INDEXXSPACE name generated will most likely be different in the source and target systems. You should query the source and target catalogs to get a list of the INDEXXSPACE names for the indexes that have a name greater than eight characters (found in SYSIBM.SYSINDEXES). These will be used in a later step.

5. On the source Db2 subsystem, run the REORG utility on any table spaces for which a table alteration has occurred with no subsequent REORG. If you are unsure if a table meets this criteria, query SYSIBM.SYSTABLES for those tables in which CREATEDTS does not equal ALTEREDTS. However, there is no way to tell from the Db2 catalog what kind of ALTER was performed and whether it would have affected this process. By comparing the CREATEDTS and ALTEREDTS, you will at least gather a list of candidates. If all rows have CREATEDTS equal to ALTEREDTS, running the REORG utility is unnecessary.

6. Move the data choosing one of the three methods that match the ones listed above:

   • **Method 1: Unload / Reload:**
     a. Start the objects on the source system in RO mode.
     b. Execute the UNLOAD utility on the source system to unload the data to a sequential file.
     c. Execute the LOAD utility on the target system to load the data that was unloaded when performing the step listed above.

   • **Method 2: DSN1COPY with OBID translation:**
     a. Stop the object, for both table space and index, on the source and target systems.
     b. Execute DSN1COPY with OBID translation and the RESET option to copy the data from the source system to the target system. If the online REORG utility has been run, be sure to code the proper I/J data set names in the DSN1COPY job. Also, if any indexes were created with names greater than eight characters long you will need to incorporate the INDEXXSPACE name differences retrieved in an earlier step.
     c. Start the objects on the target systems R/W.

   • **Method 3: Use Db2 VSAM data sets from source system in the target system:**
     a. Stop the table spaces and index spaces in the target system.
     b. If the high level qualifier (HLQ) is not changing between the source and target systems, which is not recommended, then perform the following steps on the target system. Note that this process assumes that the objects were created in the target system with DEFINE NO utility.
       1) Change SPRMCTU in the DSN6SPRM macro to one. This enables the catalog to be updated. Also, change DLDFREQ to zero, which disables the down level detection. You must note the original value, because you will change DLDFREQ back to the original value in a later step. Assemble the changed zparm and restart the target system to pick up the change.
       2) Using SQL, update the SPACE column from SYSIBM.SYSTABLEPART and SYSIBM.SYSINDEXPART from -1 to 0 for the table and index spaces that you are moving using method three.
3) Run the REPAIR DBD REBUILD utility for the databases created earlier. This step is needed because the above step changed the SPACE value in the Db2 catalog from -1 to 0, but the value is also in the DBD. This means that when the REPAIR DBD REBUILD utility is run, it will take the information from the catalog and rebuild the DBD in the directory, making Db2 think that the data set has already been defined.

**Note:** Do not try to access the tables at this point. Both systems will think that the data set is theirs, but only one system will actually be able to access the data set. The REPAIR DBD REBUILD utility only should be executed after all table spaces and index spaces that are being moved by method three for a particular database, have their catalogs updated via SQL.

4) Change SPRMCTU back to zero, and then DLDFREQ to the previous value it was assigned. Next, reassemble the zparm and recycle the target system so it recognizes the change.

c. Delete the target system VSAM data sets for the table spaces and indexes, unless they were created with DEFINE NO, because the HLQ did not change.

d. Use the REPAIR utility on the source system to change the identifiers in the page set header page, or header pages, if you have a partitioned page set, to match the new identifiers on the target subsystem. The identifiers consist of two, 2-byte fields, HPGDBID and HPGPSID. You must locate and replace these identifiers with the new DBID and PSID of the target system as follows:
   - For non-partitioned page sets, locate and replace the 4-byte fields starting at X'0C', which begins on page zero.
   - For partitioned page sets, locate and replace the 4-byte fields starting at X'0C' on page zero for each partition.

e. If the table space that you are working with is segmented and compressed, and a dictionary exists, you must also run the REPAIR utility on the source system to change the OBID in the first space map page.

f. Stop the object, for table space and index, on the source system.

g. If the HLQs of the source and target systems are different, you must run the IDCAMS ALTER statements. Running the IDCAMS ALTER statements changes the HLQ of the VSAM data sets from the source system to be the new HLQ of the target system. If the HLQ is changing, then the VSAM data sets of the target system must be first deleted before running IDCAMS ALTER. Additionally, if you know from completing one of the previous steps that the online REORG utility has been executed on the source system, make sure that you change any 'J' data sets to match the target system. The target system should have all 'I' data sets, as they were newly created. Also, if any indexes were created with names greater than eight characters long, you will need to incorporate the INDEXSPACE name differences retrieved in an earlier step.

h. Start the table spaces and index spaces in the target system.

i. Use the REPAIR utility with the LEVELID option to reset the level indicator of the page sets to a neutral value on the target system.

j. Stop the object, for table space and index, on the source system.
7. You now have the option to drop the database objects on the source subsystem. If you do not choose to drop the objects at this point, and used method three to move data while keeping the same HLQ for the data sets, then you should consider altering the VCAT name in the source system to a different invalid value. You should alter the VCAT name so that the source Db2 does not try to open the data set that is now in use by the target system.

8. Take full image copies of all data. This is the earliest time at which data recovery can occur after the merge.

9. Run the RUNSTATS utility on the target system.

10. Bind all plans and packages on the target subsystem that were bound on the source subsystem. Bind any plans and packages on the target subsystem that have changed because of name conflict resolution. Grant the appropriate authorizations to all plans and packages.

Related reference:

- [DSN1COPY (Db2 Utilities)](#)
- [REPAIR (Db2 Utilities)](#)

Running REPAIR to modify OBID for a compression dictionary:

Additional invocations of the REPAIR utility might be necessary if you use Db2 VSAM data sets from the source system to move data that you are merging in the target system and the table space is segmented and compressed.

About this task

If the segmented, compressed table space has a dictionary, the space map page that contains the dictionary has an OBID that is the same as the table space that was modified in the header page when you moved and merged your data. The OBID in the space map must be modified to match the new value in the header page.

Procedure

To modify the OBID in the space map to match the new value in the header page:

1. Use the DSN1PRNT utility to view the data set header page and space map.
   The following is an example from DSN1PRNT with the FORMAT option. On the header page, or page zero, you will see something similar to the output displayed in the following figure:

   ```plaintext
   PAGE: # 00000000 ---------------------------------------------------------------
   HEADER PAGE: PGCOMB='10'X PLOGRBA='BC8DB9277831'X PNUM='00000000'X PGFLAGS='38'X
   HGPOBID='018E0002'X HPGHPREF='000000B4'X HPGCATRL='00'X HPGREL='D3'X HPGZLD='D3'X
   ```

   Figure 87. Sample header page of DSN1PRNT command with the FORMAT option.

   The following figure shows the first space map page, or page 1:
In the header page, HPGOBID is the DBID concatenated with the OBID. In the first figure above, HPGOBID is X'018E0002', so the DBID is X'018E' and the OBID is 0002'X. You can see that same OBID in the first four segments in the space map page, or bytes 5 and 6, following the segment number.

If you run DSN1PRNT on pages 0 and 1, and you do not see the same OBID in the space map page as the one listed on the header page, it means that the dictionary is not yet built. If the dictionary is not built, you only see the table segments (segments X'5' thru X'E' shown above). If this is the case, you do not need to modify the OBID and you can stop this task.

2. Use the REPAIR utility to modify the OBIDs in the space map to match the new OBID in the header page. Only the segments that match the OBID in the old header page must be modified. The remaining segments are for the table and do not need to be modified.

In the second figure above, you can see that the first four segments need to be modified and that the table OBID is X'0003'.

The number of segments that need to be modified is dependent on the SEGSIZE value of the table space. The following table shows the offset of the OBIDs that need to be modified for each SEGSIZE.

<table>
<thead>
<tr>
<th>SEGSIZE</th>
<th>Offset of OBIDs to REPAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>X'20'</td>
</tr>
<tr>
<td></td>
<td>X'29'</td>
</tr>
<tr>
<td></td>
<td>X'32'</td>
</tr>
<tr>
<td></td>
<td>X'3B'</td>
</tr>
<tr>
<td>8</td>
<td>X'20'</td>
</tr>
<tr>
<td></td>
<td>X'2B'</td>
</tr>
<tr>
<td>12</td>
<td>X'20'</td>
</tr>
<tr>
<td></td>
<td>X'2D'</td>
</tr>
<tr>
<td>16–64</td>
<td>X'20'</td>
</tr>
</tbody>
</table>

Figure 88. Sample output from first space map page of DSN1PRNT command with FORMAT option.
Release coexistence in Db2 data sharing

In *release coexistence*, applications can continue to access Db2 data while you migrate the members of the data sharing group. However, you must weigh the benefit of continuous availability against the operational costs of running in coexistence. Most new capabilities and enhancements in Db2 12 remain unavailable during coexistence, and additional system management considerations apply.

Db2 12 supports coexistence with Db2 11 members as long as function level 500 or higher has not been activated on any Db2 12 member. Release coexistence begins when you migrate the first data sharing member to Db2 12. You must successfully migrate the first data sharing member to Db2 12 before attempting to migrate the other data sharing members.

**Important:** Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

For maximum availability, you can migrate the members to Db2 12 one member at a time. When developing your migration plan, remember that most new functions that are introduced in Db2 12 are not available to any members of the group until all members are migrated to Db2 12 and function level 500 or higher is activated.

If you do not require continuous availability, consider shutting down the group during the migration to avoid the coexistence environment. If you must run in coexistence, plan to migrate the members in as short a time as possible to minimize the operational complexity.

Plan migrations for periods of low activity because of the locks used by the catalog migration utility (CATMAINT). These locks make parts of the Db2 catalog are unavailable during migration of the first member. The locks used varies from release to release, depending on the catalog parts that require modification.

Coexistence considerations are similar to those that you need to understand for the fallback environment. For example, objects that are frozen in fallback are generally not accessible from a Db2 11 member.

### Activation of function level 500 or higher

In Db2 12, function level 500 or higher is activated when you successfully issue the ACTIVATE command on any single member of the data sharing group.

**Tip:** You can determine the catalog level and function level for a Db2 subsystem or data sharing group, and the code levels of individual subsystems or members, by issuing DISPLAY GROUP commands. For more information and examples, see [Determining the Db2 code level, catalog level, and function level](Db2 for z/OS What’s New?)

Before attempting this process, confirm that all members are running Db2 12. The ACTIVATE command is successful only if all active members are running Db2 12.
**Important:** Do not issue the ACTIVATE command or run job DSNTIJAF for activation of function level 500 or higher until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

** Important:** Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

**TSO, CAF, and RRSAF logon procedures**

You can attach to either release of Db2 with your existing TSO, CAF, or RRSAF logon procedures without changing the load libraries for your applications. After you migrate completely to the latest level of Db2, you must update those procedures and jobs to point to the latest level of Db2 load libraries. If you forget to update those procedures and jobs before migrating to any release subsequent to Db2 12, those procedures and jobs can no longer work in that subsequent release.

**Related tasks:**
- Migration step 29: Activate function level 500
- Activating Db2 function levels in a data sharing
- Preparing for Db2 12 migration in Db2 data sharing

**Related reference:**
- [ACTIVATE (Db2) (Db2 Commands)]

**Special considerations for WLM-managed stored procedures in coexistence**

The WLM-managed stored procedure address space (WLM-SPAS) that processes a stored procedure or UDF program always connects to the Db2 member that processes the SQL call request. Therefore, the Db2 code that is executed in the WLM-SPAS must match the Db2 release of the DBM1 address space, or an error occurs.

When running in a data sharing group, each member has a different DBM1 address space. This means that the Db2 code in the WLM-SPAS will have to match more than one Db2 release level. This can cause a problem when you want to use a single JCL definition for the procedure name listed in the WLM definition.

To define one JCL procedure to reference Db2 code data sets at different release levels, use a simple data set and alias naming convention. If the procedure library is shared throughout the SYSPLEX, this allows for a single JCL definition for the Procedure Name that is listed in the WLM definition.

In the following example, the DB2SSN parameter is used as a part of the data set name in STEPLIB. This allows redirection to a data set name based on the SSN of the member that is invoking the stored procedure or UDF.

In the WLM definition:
Limit on starting server address spaces for a subsystem instance:

1. No limit
2. Single address space per system
3. Single address space per Sysplex

The special keyword &IWMSSNM in the WLM definition causes WLM to send the subsystem name as the DB2SSN parm when the address space is started. In the WLM-SPAS JCL:

```plaintext
//DB2AWLM1 PROC RGN=OM, DB2SSN=, NUMTCB=1, APPLENV=DB2GROUP1
//IEFPROC EXEC PGM=DSNX9WLM, REGION=&RGN, TIME=NOLIMIT,
//   PARM=’&DB2SSN, &NUMTCB, &APPLENV’
//STEPLIB DD DISP=SHR, DSN=DSNT2.&DB2SSN..SDSNEXIT
// DD DISP=SHR, DSN=DSNT2.&DB2SSN..SDSNLOAD
// DD DISP=SHR, DSN=DSNT2.&DB2SSN..SDSNLOD2
// DD DISP=SHR, DSN=DSNT2.RUNLIB.LOAD
```

If one subsystem name is DT21 then the following alias can be used to redirect the library name to a release-specific library for that member. This would allow you to have a single release-specific library for a data sharing group.

```
DSNT2.DT21.SDSNLOAD *ALIAS
```

As members are migrated, the ALIAS can be changed to reflect the new release that the member is running.

### Determining the release of a data sharing group in coexistence

When the first data sharing member starts Db2 12, the catalog is considered to be at that level, even though not all members of the group have migrated. This means that the catalog level for the group is Db2 12.

**Procedure**

1. Issue the DISPLAY GROUP command. The catalog level is shown near the top of the display.

### Determining the function level of an IRLM group in coexistence

IRLM communicates coexistence information by using function levels. A function level is an ever-increasing number that each IRLM can use to tell other IRLMs in the group what level of function it supports.
About this task

The group function level is the minimum of the individual IRLM function levels for all IRLMs that can coexist. An IRLM that tries to join a data sharing group is prevented from doing so by active members that cannot coexist with the new IRLM’s function level.

When the function level for the group changes, that change is serialized by IRLM with lock structure rebuilds. In most cases, however, the lock structure does not actually do a full rebuild. The first phase of the rebuild is enough to quiesce the work and cause the function level change to occur. These “partial” rebuilds take place when an IRLM joins the group and if that activity causes the group function level to change. For example, if the IRLM group is currently at function level n, and the IRLM member that wants to join the group is at n-1, the partial rebuild occurs to lower the group function level.

Procedure

To determine IRLM function levels:

Enter the following command:

```
MODIFY irlmproc,STATUS,ALLI
```

Examples

Example 1: Assume a data sharing group exists with three members running Db2 Db2 12 conversion mode and one member running Db2 11. Issuing the MODIFY irlmproc,STATUS,ALLI command from one of the three Db2 12 members produces output that is similar to the following output:

```
DXR103I QR21002 STATUS 622
IRLMS PARTICIPATING IN DATA SHARING GROUP FUNCTION LEVEL=2.029
IRLM NAME IRLMID STATUS LEVEL SERVICE MIN_LEVEL MIN_SERVICE
PR21* 001 UP 3.029 PM28625 1.022 PM28625
QR21 002 UP 2.029 PM28625 1.022 PM28625
SRLM 004 UP 3.029 PM28625 1.022 PM28625
TRLM 005 UP 3.029 PM28625 1.022 PM28625
DXR103I End of display
```

The IRLMs in the preceding example are at group function level 2.029, which is the lowest level of any of the individual members (QR21). The MIN_LEVEL field shows the minimum level with which this IRLM can coexist. MIN_SERVICE indicates the service or release that corresponds with that MIN_LEVEL.

Example 2: Assume the same data sharing group as described in the previous example. Issuing the MODIFY irlmproc, STATUS, ALLI command from the Db2 11 member produces output that is similar to the following output:
Call attachment and TSO attachment in coexistence

While operating in a coexistence environment, you can attach to either Db2 release with your existing TSO logon procedures or with JCL.

After you migrate all members of the group to Db2 12, update those procedures and jobs to point to the Db2 12 load libraries.

Automatic binds in coexistence

Changes to package structures that are introduced in Db2 12 are not available to members of the group that have not migrated. You must plan accordingly when developing your migration plan.

Important: To reduce coexistence problems, rebind all packages and plans that can run in Db2 11, but cannot run without an automatic rebind in Db2 12. To identify those packages, run job DSNTIJPM, and examine the premigration reports on package copies and plans that are not supported after migration to Db2 12. For details, see [Run premigration queries (DSNTIJPM)](https://www.ibm.com/docs/en/db2/12.1?topic=run-premigration-queries-dsntijpm).

Important: Apply APAR PI87675 to eliminate the possibility of repeating automatic remigration binds during release coexistence.

Migration-related automatic binds occur in Db2 12 because it cannot use runtime structures from a plan or package that was last bound in a release earlier than DB2 10. Plans and packages that were bound in Db2 11 can run in Db2 12, without the risk of migration-related automatic binds. However, plans and packages that are bound in Db2 12 cannot run on Db2 11 members without an automatic bind in Db2 11.

If automatic binds are allowed to occur, they are inevitably disruptive for rolling online migrations, or any time that different releases coexist in data sharing environments. During and after migration, such automatic binds can result in costly problems, including performance regression and even application outages.

Automatic binds occur the first time that any plan or package runs in Db2 12, which means that the automatic binds, and resulting problems and application outages, can occur well after a migration window. That is, such problems and resulting outages can occur any time in the “first impression” window in Db2 12, which is any time until the plan or package first runs in Db2 12.

The best approach is to bind all packages and plans that might be subject to migration-related automatic binds in Db2 11 with good performance before you start the migration to Db2 12. Doing so reduces the risk of the following problems that can result from automatic binds:
• Application failures because of contention between releases. For any migration that brings into release coexistence, automatic binds might fail for plans or packages that are in use by a Db2 11 member, causing applications to fail on the Db2 12 member.

Successful automatic binds on Db2 12 in a coexistence environment also create a package or plan that Db2 11 members cannot run without another automatic bind. The automatic bind might also fail if the plan or package is in use by a Db2 12 member.

• Application failures, when automatic binds fail for access control authorization exit users because the authorization ID for the automatic bind has insufficient privileges, compared to the owner of the package or the plan.

• Difficult to resolve performance regressions. For packages, a migration-related automatic bind also destroys the current package. If the automatic rebind succeeds but introduces an access path regression, REBIND SWITCH is not possible. Any performance regression that occurs in Db2 12 might also occur in Db2 11.

For more information about avoiding automatic binds during or after the Db2 12 migration process, see Avoid automatic binds at migration to Db2 12.

The migration process binds new packages in Db2 12, including packages for Db2-supplied stored procedures, user defined functions, and tools such as SPUFI and the Db2 REXX language support. Data sharing members in the Db2 12 load and execute those programs from the Db2 12 SDSNLOAD library. Data sharing members on Db2 11 continue to load this packages from the SNDSNLOAD library for Db2 11.

If you must rebind some of your application packages in Db2 12 while coexistence with Db2 11 continues, you must consider how to handle the resulting binds and automatic rebinds while the different releases coexist.

Related concepts:
- Automatic rebinds (Db2 Application programming and SQL)
- Release coexistence in Db2 data sharing

Related tasks:
- Avoid automatic binds at migration to Db2 12

Related reference:
- AUTO BIND field (ABIND subsystem parameter)

Recommendations for BIND in coexistence

If DSN is at Db2 12 and the member that is named in the DSN command is at Db2 11, using certain bind options causes a BIND or REBIND subcommand to be rejected.

If you are migrating from Db2 11, the following options cause BIND and REBIND commands to be rejected:
- APPLCOMPAT(V12R1 | V12R1M500 | V12R1M100)
- APREUSESOURCE(PREVIOUS | ORIGINAL | CURRENT)
- CONCENTRATESTMT(YES | NO)

To avoid problems, make sure that the member that is named in the DSN subcommand matches the load libraries that are used for the DSN command.

Related concepts:
Automatic binds in coexistence

Related tasks:
- Avoid automatic binds at migration to Db2 12

Related reference:
- DSN (TSO) (Db2 Commands)
- BIND PACKAGE (DSN) (Db2 Commands)
- REBIND PACKAGE (DSN) (Db2 Commands)

Recommendations for utilities in coexistence

Until all members of the data sharing group are running Db2 12, avoid using any of the new utility functions that are available in Db2 12. However, as long as you use utility options that are supported in Db2 11, utilities can attach to either a Db2 11 or Db2 12 member.

**Important:** In data sharing release coexistence, you must restart a utility on a member on the same Db2 release as the member where the utility job was originally submitted. The same utility ID (UID) must be used to restart the utility. That UID is unique within a data sharing group. However, if Db2 fails, you must restart Db2 on either the same or another z/OS system before you restart the utility.

The utilities batch program (DSNUTILB) is split into multiple load modules: a release-independent load module called DSNUTILB, multiple release-dependent modules DSNUT111 or DSNUT121, and, for Db2 11, multiple utility-dependent load modules. To operate in a mixed-release data sharing environment, you must have DSNUT111 (if applicable), and DSNUT121. And you must have all utility-dependent load modules and their aliases for the utilities that you have purchased available to the utility jobs that operate across the data sharing group. The following examples show how you can make these load modules available.

### Changing STEPLIB in DSNUPROC

The recommended method for making the release-dependent modules available for utility jobs is to change the STEPLIB in DSNUPROC to include the other release, as in the following example:

```plaintext
//DSNUPROC PROC LIB='DSN1210.SDSNLOAD', // SYSTEM=DSN, // SIZE=0K,UID='',UTPROC=' //DSNUPROC EXEC PGM=DSNUTILB,REGION=&SIZE, // PARM='&SYSTEM,&UID,&UTPROC' //STEPLIB DD DSN=DSN1110.SDSNLOAD,DISP=SHR; // SYSPLIB DD DSN=DSN1110.SDSNLOAD,DISP=SHR ← coexistence //SYSPRINT DD SYSOUT** //UTPRINT DD SYSOUT** //SYSUOMP DD SYSOUT** //DSNUPROC PEND REMOVE * FOR USE AS INSTREAM PROCEDURE
```

### Cross-copy into load libraries

Another approach, which is not recommended for long-term use, is to cross-copy the release-dependent modules into the load libraries of the other release. For example, copy DSNUT111 into the Db2 12 load libraries, and copy DSNUT121 and all applicable utility-dependent load modules into the Db2 11 load libraries. The problem with this approach is that you must repeat this procedure every time you...
apply maintenance to these modules. Thus, as with coexistence in general, this approach is only for short-term use.

The following shows sample JCL to perform the cross-copy:

```jcl
//
CROSCOPY PROC D121TPRE='DSN1210',
// D111TPRE='DSN1110',
// RGN=4096K,SOUT='*'
/*/ FOR EXECUTION OF IEBCOPY - DB2 POST-INSTALLATION ***
/*/******************************************************************************
COPY EXEC PGM=IEBCOPY,REGION=&RGN
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
/*/ ****************** DB2 TARGET LIBRARIES __________________________
/*/ /D111LOAD DD DSN=&D111TPRE..SDSNLOAD,DISP=OLD
/D121LOAD DD DSN=&D121TPRE..SDSNLOAD,DISP=OLD
/*/ PEND
//COEXIST EXEC PROC=DSNTIJCO
//SYSIN DD *
COPYMOD INDD=((D121LOAD,R)),OUTDD=D111LOAD
SELECT MEMBER=(DSNU121)
COPYMOD INDD=((D111LOAD,R)),OUTDD=D121LOAD
SELECT MEMBER=(DSNU111)  <-- add any utility-dependent modules here
```

**Related information:**

**Recommendation for group restart in coexistence**

If a group restart is necessary while the data sharing group is running in a coexistence environment, issue the START command only for Db2 12 members.

Do not start the Db2 11 members until the Db2 12 members have completed forward log recovery. If a Db2 11 member performs the group restart for a Db2 12 member, Db2 11 adds pages to the logical page list during the peer-forward recovery phase when it tries to apply redo log records against a release-dependent object.

**Recommendation for SPUFI in coexistence**

When you migrate the first member of the data sharing group to Db2 12, you run job DSNTIJSG, which binds the SPUFI packages in Db2 12.

Binding SPUFI in Db2 12 causes SPUFI to be unavailable to the Db2 11 members. If you attempt to run an SQL statement on a member that has yet to migrate to Db2 12, expect messages that indicate an unavailable resource.

**IRLM service and release coexistence**

As you apply IRLM service to members of a data sharing group, some members run with the newer service level, and some run with the older service level.

A mix of service levels can raise issues that you must consider.

When the IRLM subsystems in the group are at different service levels, the group function level for IRLM is the lowest IRLM function level. All IRLM subsystems in the group are functional, but any enhancements or fixes that were made at the higher function levels are not available.
Preparing for Db2 12 migration in Db2 data sharing

When planning the migration of data sharing members to Db2 12 it is best to take certain actions to reduce or avoid unnecessary operation complexity.

Procedure

To prepare for migration to Db2 12 in data sharing, complete the following steps:

- Ensure that all required maintenance, including the Db2 12 fallback SPE, is applied on every member.

**Important:** Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

- After all members have migrated to Db2 12, do not activate new function until your data sharing group has processed a full range of typical work. The amount of time that a data sharing group needs before new function is activated depends on the size of the data sharing group and the complexity of its typical work.

**Important:** Do not issue the ACTIVATE command or run job DSNTIJAF for activation of function level 500 or higher until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

- Before you migrate to Db2 12, read the information about release incompatibilities and make any necessary changes. For more information, see [Incompatible changes in Db2 12 (Db2 for z/OS What’s New?)](#).

- Plan to minimize the amount of time that Db2 12 operates in release coexistence with members on Db2 11. For more information about release coexistence in Db2 data sharing, see [Release coexistence in Db2 data sharing](#).

- Start only one Db2 member for migration processing. During the migration, other group members can be active. However, other active group members may experience delays or timeouts if they attempt to access catalog objects that are locked by migration processing. After migration completes on the first member, you can migrate the other data sharing group members.

- To prepare for possible fallback from Db2 12, keep the subsystem parameter load module that is used by Db2 11.

Related concepts:

- Planning for Db2 data sharing
- Release coexistence in Db2 data sharing

Related tasks:

- Installing, migrating, and enabling Db2 data sharing
- “Migration step 29: Activate function level 500 or higher” on page 538
- Migrating an existing data sharing group to Db2 12
Migrating an existing data sharing group to Db2 12

If you have a Db2 11 Db2 data sharing group, and you want a Db2 12 data sharing group, you can migrate to the newer release.

Before you begin

Prepare your data sharing group for migration, as described in Preparing for Db2 12 migration in Db2 data sharing.

During the migration to Db2 12, other group members can be active, but they might experience delays or timeouts if they attempt to access catalog objects that are being updated or that are locked by migration processing. Before beginning the migration process, confirm that Db2 11 is at the proper service level. Also, check your coupling facilities to ensure that the appropriate service levels are installed. Having the wrong service levels installed can result in data corruption. No specific service level requirements exist for CFLEVELs other than 12, but CF levels 13 and 14 are recommended for additional function. Use the z/OS D CF command to display the service levels for IBM coupling facilities.

Important: Migration to Db2 12 is only supported from Db2 11.

Important: If you are migrating the source, proxy, and target data sharing groups in a GDPS Continuous Availability with zero data loss environment to Db2 12, see Upgrading a Db2 11 GDPS Continuous Availability with zero data loss environment to a Db2 12 environment (Db2 Administration Guide) for the order in which you need to perform the migrations.

Important: Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

Before you start to migrate your data sharing group:

- Read the information about migration considerations in Incompatible changes in Db2 12 (Db2 for z/OS What's New?)
- Read the information in 00C10207 (Db2 Codes)

Related concepts:
The scope and uniqueness of Db2 subsystem parameters in data sharing

Migrating the first member of a group to Db2 12

You must first complete the migration of one member of the data sharing group to Db2 12 before starting any other members at Db2 12. You may choose any member as the first member to migrate to the new release.
About this task

The DB2I CLISTs are edited when the first member of a data sharing group migrates to Db2 12. The sample jobs are edited when the first member migrates to Db2 12, and they are also edited as part of preparation for enabling-new-function mode for the group.

Jobs that the migration CLIST tailors: Jobs that are marked with an asterisk (*) are modified only when you migrate the first member of a data sharing group.

<table>
<thead>
<tr>
<th>Job</th>
<th>Modified</th>
<th>Modified</th>
<th>Modified</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIJCB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJJCX*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJEX*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJIC*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJGD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJJC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJJE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJKH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJMW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJPD*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJPU</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJQC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJRT*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJRV*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNTIJUZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Follow these directions carefully. The first member of the data sharing group uses DSNTIDXA as its input member name. If you use SAP, specify DSNTIDXB, the SAP-specific CLIST input member, instead of DSNTIDXA. A subsequent member must use a previous member’s output member name as its input member name.

Procedure

To migrate the first member of a data sharing group to Db2 12:

1. Make a plan to migrate the data sharing group in as short a time period as possible. Migrating an existing data sharing group requires careful planning.
2. Apply maintenance through the Db2 12 fallback SPE to all active members.

   **Important:** Apply the fallback SPE and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member.

   In case fallback becomes necessary, keep the subsystem parameter load module that is used by Db2 11.

3. For the first member to migrate, specify the following settings on installation panel DSNTIPA1:

   INSTALL TYPE ===> MIGRATE
   DATA SHARING ===> YES

   DATA SET(MEMBER)NAME ===> 

   INPUT MEMBER NAME ===> DSNTIDXA or DSNTIDXB
   OUTPUT MEMBER NAME ===> this member’s output PDS member

4. On installation panel DSNTIP0B, specify 1 to indicate that this is the first member of the group to migrate.
5. Complete the installation panels. Specify parameters according to the Db2 subsystem parameters guidelines.
6. Complete all of the **steps to migrate to Db2 12**

Related concepts:

Subsystem parameters and application defaults in a data sharing environment
Migrating subsequent members of a group to Db2 12

After you complete the migration of one member of a data sharing group to Db2 12, you can migrate subsequent members of the group.

About this task

To migrate subsequent members of a group to Db2 12, complete these tasks for each member:

**Related tasks:**

- Starting inactive data sharing group members in a new Db2 release

**Running the CLIST for migrating data sharing members**

The process of migrating data sharing members to Db2 12 involves running some of the migration jobs. Before you run the migration jobs, tailor them to your environment by running the installation CLIST.

Before you begin

Be sure to complete the process of migrating the first member of the group before you begin to migrate subsequent members to Db2 12.

Procedure

To tailor the installation jobs for migrating data sharing members:

1. Invoke the installation CLIST as described in Tailoring Db2 12 installation and migration jobs with the CLIST.
2. On panel DSNTIPA1, specify the following fields.
   - INSTALL TYPE ==> MIGRATE
   - DATA SHARING ==> YES
   - DATA SET(MEMBER)NAME ==> 
   - INPUT MEMBER NAME ==> first member's output PDS member
   - OUTPUT MEMBER NAME ==> this member's output PDS member
3. Specify 2 on installation panel DSNTIP0B to indicate that this is not the first member of the group to migrate.
4. Complete the installation panels, specifying parameters according to the Db2 subsystem parameters guidelines.

**Related concepts:**

- Subsystem parameters and application defaults in a data sharing environment
- The Db2 installation CLIST panel session

**Connecting Db2 to IMS (optional)**

Connecting Db2 to IMS is an optional part of migrating existing data sharing members to a new release. Connecting Db2 to IMS requires coordination with your company's group that supports IMS systems.

Procedure

To connect the IMS attachment facility:

1. Make Db2 load modules available to IMS
2. Define Db2 to IMS
3. Define new programs and transactions to IMS
4. Prepare IMS applications for Db2
5. Depending on your site, you might also need to:
   • Define Db2 plans for IMS applications
   • Generate a user language interface

**Related tasks:**
Connecting the IMS attachment facility

**Connecting CICS to Db2 (optional)**
Connecting CICS to Db2 is an optional part of migrating members in a data sharing group. Connecting Db2 to CICS requires that you regenerate several CICS tables with additional entries.

**About this task**
A macro is supplied with CICS to define the connection between CICS and Db2 by using a resource control table (RCT).

Ensure that you coordinate the attachment facility connection with your CICS support group.

**Procedure**
To connect the CICS attachment facility:
1. Recalculate space requirements for the CICS attachment facility.
2. Define your CICS attachment facility parameters using the RCT.
3. Update the CICS system tables.
4. Update the CICS initialization JCL.
5. Coordinate Db2 and CICS security if necessary.
6. Prepare new CICS applications for Db2 if necessary.

**Related tasks:**
Defining the CICS Db2 connection

**Stopping Db2 activity**
Before running any installation jobs to migrate a data sharing member, ensure that all work is stopped on the subsystem.

**Procedure**
To stop work on Db2:
1. Issue the following command, where `-DSN1` is the subsystem command prefix that is defined for Db2. You must have system administrator (SYSADM) or system operator (SYSOPR) authority to issue this command.

   ```
   -DSN1 STOP DB2 MODE(QUIESCE)
   ```

   The QUIESCE keyword allows Db2 to complete the processing of programs that are currently executing. This might require some processing time.
2. Issue the following command to allow only the system administrators and system operators that were defined during installation to access Db2:

```
-Dsn1 start db2 access(maint)
```

If Db2 does not start properly, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. For example, if the bootstrap data set (BSDS) does not match the subsystem parameter values, ensure that the correct jobs were run for DSNTIJUZ. Ensure that you started Db2 with the correct subsystem parameter option.

3. Follow these steps to ensure that all Db2 11 work is stopped:
   - Make sure no units of recovery remain. Issue the following command:
     ```
     -dsn1 display thread(*) type(*)
     ```
     Then use -dsn1 recover indoubt for any indoubt threads.
   - Ensure that all utilities are completed or terminated. To ensure that no utility work remains, issue the following command:
     ```
     -dsn1 display utility(*)
     ```
     After you have determined the utilities that are running, you can let them complete processing or you can terminate the utility. To stop all utilities, issue the following command:
     ```
     -dsn1 term utility(*)
     ```
   - Ensure that no table spaces and index spaces in the Db2 directory (DSNDB01) or the Db2 catalog (DSNDB06) have write error ranges or deferred restart states. To determine existing restrictions, issue the following commands:
     ```
     -dsn1 display database(DSNDB01) spacenam(*) restrict
     -dsn1 display database(DSNDB06) spacenam(*) restrict
     ```
     You must have system administrator or system operator authority (specified on installation panel DSNTIPP1) to issue this command.
     Recover any table spaces and index spaces that have write error ranges or deferred restart states by using the RECOVER utility.
4. Stop Db2 by issuing the following command:

    -DSN1 STOP DB2 MODE(QUIESCE)

A user with installation system administrator (installation SYSADM) or installation system operator (installation SYSOPR) authority must enter this command.

Related reference:
- DISPLAY DATABASE (Db2) (Db2 Commands)
- DISPLAY THREAD (Db2) (Db2 Commands)
- DISPLAY UTILITY (Db2) (Db2 Commands)
- STOP DB2 (Db2) (Db2 Commands)
- TERM UTILITY (Db2) (Db2 Commands)

**Defining Db2 12 initialization parameters for migrating data sharing members**

For migrating data sharing members after the first to Db2 12, you must define Db2 12 initialization parameters in the subsystem parameter module (DSNZPxxx) and update DDF related information in the BSDS.

**About this task**

The subsystem parameter module (DSNZPxxx) and application defaults module (dsnhdecp) names are specified in the following DSNITPO3 installation panel fields:

- **PARAMETER MODULE field**
- **APPL DEFAULTS field**

For migrating members, the application defaults load module (dsnhdecp) is not reassembled for each subsequent migrating member.

**Procedure**

To define the Db2 12 initialization parameters, complete the following steps:

Run job DSNTIJUZ. Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of prefix.SDSNEXIT and prefix.SDSNLOAD, you can move the SYSLMOD output to that library.
- If you changed the prefix for the Db2 distribution libraries, edit DSNTIJUZ to correct the data set names.
- If you have not run the SMP/E ACCEPT job (DSNACEP1) of FMID HDBCC10, edit DSNTIJUZ so that the SMP/E temporary data set (SMPTLIB) is included in the concatenation for the ADSNLOAD DD statement in step DSNTIZL. This action ensures that members DSNFSYSP, DSNJARVP, DSNJLOGP, DSNTPRM,
DSNVDIR1, DSNZMSTR, and DSN3DIR1 are linked with the subsystem parameter load module. SMPTLIB is hlq.HDBCC10.F2, where hlq is from the GLOBAL SMP/E zone. Use the following SMP/E statements to get DSPREFIX:

```
SET BOUNDARY (GLOBAL).
LIST DDDEF (SMPTLIB).
```

Insert the DSPREFIX value after SDSNLOAD and ADSNLOAD.

When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions. For more information, see [job DSNTIJUZ: define the Db2 data-only subsystem parameter module](#).

### Results

The DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions.

- [Subsystem parameters (Introduction to Db2 for z/OS)](#)
- [Application programming default values (Introduction to Db2 for z/OS)](#)
- Directory of subsystem parameters, panel fields, and application default values
- [Bootstrap data set (Introduction to Db2 for z/OS)](#)

#### Establishing subsystem security: DSNTIJRA (optional)

Db2 lets you control access to data within Db2. Db2 also works together with outside security systems, such as RACF, that control access to the Db2 subsystem.

### About this task

Because your Db2 12 subsystem reuses the data objects from your Db2 11 subsystem, you have probably already supplied the protection that those objects need. However, you probably want to protect the new (Db2 12) Db2 data objects.

### Procedure

To establish subsystem security:

1. Customize job DSNTIJRA, as described in the job prolog.
2. Run job DSNTIJRA. Job DSNTIJRA defines the administrative task scheduler started task module to RACF program control and defines the administrative task scheduler as a trusted context in RACF. DSNTIJRA is intended for the site security administrator, and can be run any time prior to installing or migrating to Db2 12.
3. If you have a Db2 data sharing environment, customize and run DSNTIJRA for each member of the group.

### Related concepts:

- [Getting started with Db2 security (Managing Security)](#)

#### Updating Db2 address space definitions for data sharing members to z/OS: DSNTIJMA

For data sharing members after the first, you must update the z/OS system procedure libraries for the Db2 12 address spaces.
Before you begin

The authorization ID that runs the jobs must have authority to update the target system PROCLIB. You can enter that library name on installation panel DSNTIPW in the SYSTEM PROCEDURES field. You might want to use a procedure library other than SYS1.PROCLIB for the procedures.

Examine the updates from job DSNTIJMA carefully before you run it. Also, you might have additional procedures of your own to rename.

Procedure

To update the Db2 address space definitions to z/OS:

Tailor and run job DSNTIJMA. Job DSNTIJMA renames and replaces the startup procedures in the target system procedure library for the following Db2 address spaces, where ssnm is the subsystem name:
- ssnmMSTR for Db2 system services
- ssnmDBM1 for Db2 database services
- ssnmDIST for the Db2 distributed data facility (DDF)
- ssnmADMT for the Db2 administrative task scheduler
- IRLMPROC, or a user-defined procedure for IRLM

Examine the size of the private area on the Db2 startup procedures. If necessary, modify the procedures to satisfy the requirements for environmental descriptor manager (EDM) pool size, buffers, number of data sets open, and amount of available private address space. For more information, see Calculating working storage requirements.

The STEPLIB concatenation of the ssnmDBM1 address space procedure includes a commented-out DD for the IBM Language Environment data sets SCEERUN and SCEERUN2. If your system does not include SCEERUN and SCEERUN2 in the system link list, you must uncomment this DD.

Starting Db2 12

During migration to Db2 12, start the Db2 subsystem after you run job DSNTIJMV and IPL z/OS.

Procedure

To start Db2:

1. Start the IRLM. If you have not requested that Db2 automatically start the IRLM, you should start it before you start Db2. Use the following command, where irlmproc is the name that you assigned to the IRLM startup procedure:
   ```
   START irlmproc
   ```
   The irlmproc is the value that you specified for the PROC NAME option on installation panel DSNTIPI.
   If you specified YES for the AUTO START option on installation panel DSNTIPI, Db2 starts the IRLM automatically.

2. Start Db2 from the z/OS console with the following command, where -DSN1 is the subsystem command prefix that you defined for Db2, and DSNZPxxx is the name of the Db2 initialization parameter module:
   ```
   -DSN1 START DB2 PARM(DSNZPxxx)
   ```
If you omit the PARM parameter, and you specified a value in the PARAMETER MODULE field on installation panel DSNTIP03, that value is used as the parameter module name. If you did not specify a parameter module on panel DSNTIP03, Db2 uses the default, DSNZPARM.

If you did not run job DSNTIJRT when you migrated the first member of the group and the administrative task scheduler starts, the administrative task scheduler issues error message DSNA679I to report SQLCODE -805 until you run job DSNTIJRT. The administrative task scheduler starts if you specified an address space procedure name in the ADMIN SCHEDULER field on installation panel DSNTIPX.

If Db2 starts successfully, two to five address spaces also start. These address spaces are ssnmMSTR and ssnmDBM1, possibly ssnmDIST, possibly ssnmADMT, and irlmproc, where ssnm is the Db2 subsystem name, ssnmADMT is the Db2 administrative task scheduler procedure name, and irlmproc is the IRLM procedure name.

If Db2 starts successfully, the series of RESTART messages that you receive concludes with these two messages:

DSNR002I  RESTART COMPLETED
DSN9022I  DSNYASC 'DSN1 START DB2' NORMAL COMPLETION

If Db2 does not start properly, it usually abends with a reason code that indicates where the error occurred. To find the error, check the set of definitions for the associated resource. A common cause of startup failure is that the BSDS does not match the subsystem parameter values; ensure that the correct job was run for DSNTIJUZ. Also, check that the subsystem parameter member that you specified (or used by default) when you started Db2 is the one that the DSNTIJUZ job built. Check the JCL for the Db2 startup procedure.

3. If you want to use the TSO SUBMIT command to do housekeeping and migration verification, start TSO (if it is not already started).

4. If you did not enable primary user IDs to issue Db2 commands from the z/OS console or TSO SDSF when you first installed the subsystem, take one of the following actions:

   • Grant SYSOPR authority to all primary user IDs that issue Db2 commands from the z/OS console or TSO SDSF. Issue the following command:
     
     ```
     GRANT SYSOPR TO userid
     ```

   • Define RACF classes to authorize Db2 commands. Use the following statements:

     ```
     SETR CLASSACT(DSNADM)
     RDEFINE DSNADM DSN1.SYSOPR UACC(NONE)
     SETR RACLIST(DSNADM) REFRESH
     PERMIT DSN1.SYSOPR CLASS(DSNADM) ID(userid) ACCESS(READ)
     SETR RACLIST(DSNADM) REFRESH
     ```

Related reference:

- [START DB2 (Db2) (Db2 Commands)]
- [START irlmproc (z/OS IRLM) (Db2 Commands)]

Creating additional work file table spaces (optional)

During migration, you have the option of specifying on installation panel DSNTIP9 that additional work file table spaces are to be created for that member.

Procedure

If you specified that one or more additional work file table spaces are to be created for this member:
Run job DSNTIJTM.

**Verifying your Db2 12 conversion mode system (optional)**

Verify your Db2 12 conversion mode system by running some of the sample jobs from the previous release.

**About this task**

If all of the local Db2 objects from Db2 11 still exist (that is, if you have not run job DSNTJEJ0), follow this procedure.

**Procedure**

To verify the migration of IVP objects in your Db2 12 system:

1. Change the JOBLIB statements to point to `prefix. SDSNLOAD`.
2. Ensure that the DSN8EAE1 module that you created when you originally ran the Db2 11 sample jobs is copied to `prefix.SDSNEXIT`. DSN8EAE1 is an EDITPROC that is used by the employee sample table.
3. Edit the Db2 11 sample jobs before running them. Do not run all the sample jobs. Run only the specific jobs and job steps that are listed in the following steps.
4. Test the migration of the IVP phase 2 applications from your previous version:
   a. DSNTJEJ2A: Perform all except the first two steps of job DSNTJEJ2A. Expect a return code of 4 because table spaces DSN8D11U.NEWDEPT and DSN8D11U.NEWPHONE are placed in COPY-pending states.
   b. DSNTJEJ2C: Issue only the RUN PROGRAM(DSN8BC3) PLAN(DSN8BH11) statement in step PH02CS04.
   c. DSNTJEJ2D: Issue only the RUN PROGRAM(DSN8BD3) PLAN(DSN8BD11) statement in step PH02DS03.
   d. DSNTJEJ2E: Issue only the RUN PROGRAM(DSN8BE3) PLAN(DSN8BE11) statement in step PH02ES04.
   e. DSNTJEJ2F: Issue only the RUN PROGRAM(DSN8BF3) PLAN(DSN8BF11) statement in step PH02FS03.
   f. DSNTJEJ2P: Run only step PH02PS05.
5. Test the migration of the IVP phase 3 applications from your previous version:
   a. Do not run job DSNTJEJ3C or DSNTJEJ3P.
   b. If you want to test the Db2 11 ISPF-CAF applications under Db2 12, place the Db2 11 SDSNSPFP panel library ahead of the Db2 12 SDSNSPFP panel library in the ISPPLIB concatenation. This placement is necessary so that the plans that are migrated from Db2 11 can be used. Remove the Db2 11 SDSNSPFP library from your ISPPLIB concatenation when you are finished testing the Db2 11 IVP applications under Db2 12.

**What to do next**

Do not run any other Db2 11 sample jobs.

**Related tasks:**

- Ensure that sample objects from your previous release are available
Falling back and remigrating in a data sharing environment

In a data sharing environment, you can fall back from Db2 12 or remigrate one member at a time or all members at the same time.

**Important:** Fallback becomes impossible for any members of a data sharing group as soon as the -ACTIVATE command completes successfully to activate function level 500 or higher for any member of the data sharing group.

**Related concepts:**
- Release coexistence in Db2 data sharing

**Related tasks:**
- Migration step 29: Activate function level 500

**Related reference:**
- [ACTIVATE (Db2) (Db2 Commands)](https://www.ibm.com/support/knowledgecenter/SSYJH1_12.1.0/com.ibm.db2.luw.sql.ref.doc/doc/c0072587.html)

Falling back in a data sharing environment

If you need to fall back to Db2 11, you can fall back one member of the data sharing group at a time. Other members can continue to run while one member is falling back.

**Before you begin**

**Important:** Do not issue the ACTIVATE command or run job DSNTIJAF for activation of function level 500 or higher until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

If all members have already been migrated to the new release, and you fall back one member at a time, the group is running in coexistence mode after the first member falls back.

**Procedure**

To fall back to Db2 11:

1. Stop Db2 on the member that is falling back or stop Db2 on all members if you are falling back all members at once.
2. Run job DSNTIJFV to reactivate Db2 11 for that member or for all members.
3. Reconnect TSO, IMS, CICS to Db2 11.
4. Start Db2 11:
   a. Enter the command **START DB2**:
      1) Check for indoubt units of recovery.
      2) Check for outstanding restrictive states.
5. Verify Db2 11.
6. Repeat steps 1 through 5 for each member (or all members) of the data sharing group.

**Related concepts:**
- Release coexistence in Db2 data sharing

**Related tasks:**
- Creating an automatic restart policy
Remigrating in a data sharing environment

You can remigrate one member at a time or all members of a data sharing group.

Before you begin

Before you remigrate, refer to [Incompatible changes in Db2 12 (Db2 for z/OS) What’s New?](#) because many of those considerations apply to remigrations, too. Whether each consideration applies to your subsystem depends on the type of activity that took place on your Db2 11 subsystem after falling back.

Regardless of the ABIND subsystem parameter value, plans and packages continue to run on Db2 12 as bound on Db2 11, until you explicitly rebind them on Db2 12. It is best to do so only after you are certain that Db2 12 is stable and release coexistence and fallback are no longer a concern.

**Recommendation:** Avoid automatic rebinds during migration and remigration. After your remigrated system is stable, you can rebind your packages with PLANMGMT(EXTENDED). If a new package does not perform adequately, you can rebind with SWITCH(PREVIOUS) to return to the old package.

Procedure

To remigrate your subsystem to Db2 12:

1. Check the integrity of your Db2 11 catalog and directory:
   - Run the DSN1COPY utility with the CHECK option on your catalog table spaces.
   - Run the CHECK DATA utility.
   - Run the queries in member DSNTESQ of `prefix.SDNSAMP`.

2. Optional: Ensure that your Db2 11 DSNTIJIC job has been modified to copy new Db2 12 catalog table spaces, and then use DSNTIJIC to take an image copy of your Db2 11 catalog. This step is not required, but it is recommended.

3. Stop Db2 11.

4. Reconnect TSO, IMS, and CICS to Db2. Re-establish your Db2 12 logon procedures and JCL, as well as your Db2 12 CICS and IMS connections.

5. Rebuild Db2 12 cataloged procedures. Rename the Db2 12 procedures that were renamed by job DSNTIJFV during fallback. If job DSNTIJFV was not run, you need to rerun job DSNTIJMV. Comment out step 1 (DSNTIIMP), which defines Db2 12 to z/OS, and run the job. (You do not need to define Db2 12 to z/OS a second time.)

6. Start Db2 12. Ensure that you are using your Db2 12 subsystem parameter load module.

7. Take an image copy of the Db2 12 catalog by using the DSNTIJIC job.

8. Verify your Db2 12 system.

**Related concepts:**

- Release coexistence in Db2 data sharing

**Related tasks:**

- Falling back in a data sharing environment
- Re-enabling Db2 data sharing
Activating Db2 function levels in a data sharing

After all active members in the data sharing group are running Db2 12, you can activate function level 500 or higher, to enable the use of new Db2 12 capabilities.

Before you begin

Ensure that all active members are running on Db2 12. For instructions, see Migrating an existing data sharing group to Db2 12

Important: If you are activating function level 500 or higher for the first time, do not start this task before you read Migration step 29: Activate function level 500

About this task

Activating a Db2 function level for any single group member activates that function level on all members.

Procedure

To activate Db2 function levels in data sharing groups:

Complete one of the following tasks:

- Migration step 29: Activate function level 500
- Activating Db2 12 function levels (Db2 for z/OS What's New?)
- Activating Db2 function levels by using z/OSMF (Db2 for z/OS What's New?)

Related tasks:

- Adopting new capabilities in Db2 12 continuous delivery (Db2 for z/OS What's New?)

Related reference:

- What's new in Db2 12 function levels (Db2 for z/OS What's New?)

Starting inactive data sharing group members in a new Db2 release

After migration to Db2 12, you can activate data sharing group members that were last started in previous Db2 releases.

About this task

When you activate function level 500 or higher in a data sharing group, all active group members must be running on Db2 12.

Procedure

To start data sharing members that were last started in a previous Db2 release:

1. If the inactive member was deactivated, reactivate it by following the instructions in Restoring deactivated data sharing members (Db2 Data Sharing Planning and Administration)
2. Migrate the inactive member to Db2 12 by following the instructions in Migrating subsequent members of a group to Db2 12
3. If the member was last started in a Db2 release before DB2 9, run the DSNJCNVB utility to convert the BSDS of the inactive member. If the inactive
member was never started in the data sharing group after DB2 Version 7, you must use the DSNJCNVB job from a DB2 9 or a newer release to complete this step.

4. If the BSDS of the inactive data sharing member has not been converted to use the expanded RBA and LRSN fields, run the DSNJCNVT utility complete the conversion. You can use the DSNTIJCB sample job to run the DSNJCNVT utility. You must run the DSNJCNVB utility before you complete this step.

If the member was last started in a release earlier than Db2 11, the BSDS conversion must be completed. You can use the DSNJU004 (print log map) utility to determine whether the BSDS for that member has already been converted to support the extended format. If the output contains the following message, the BSDS conversion is required: DSNJCNVT CONVERSION PROGRAM HAS NOT RUN DDNAME=SYSUTx

5. Ensure that the member has at least the minimum required catalog level and code level for the data sharing group. You can use the DISPLAY GROUP command to determine the required code and catalog levels. For more information about code and catalog levels, see What's new in Db2 12 function levels (Db2 for z/OS What's New?)

6. Start the inactive member.

7. Restart active members so that they can record the updated status of the reactivated member.

Related concepts:
- Starting a data sharing member (Db2 Data Sharing Planning and Administration)

Related tasks:
- Deactivating data sharing members (Db2 Data Sharing Planning and Administration)

Convert the BSDS, Db2 catalog, and directory to 10-byte RBA and LRSN format

Related reference:
- DSNJCNVB (Db2 Utilities)
- Job DSNTIJCB
- DSNJCNVT (Db2 Utilities)
- START DB2 (Db2) (Db2 Commands)
- DSNJU004 (print log map) (Db2 Utilities)

Job DSNTIJCB

Job DSNTIJCB supports the DSNJCNVT stand-alone utility for converting the BSDS records to support 10-byte RBA and LRSN fields.

DSNTIJCB consists of the following job steps:

DSNTDEF
Activates IDCAMS to define new BSDS data sets with CLUSTER, DATA, and INDEX components under the temporary names prefix.BSDS01.NEW and prefix.BSDS02.NEW.

DSNTCNVT
Activates DSNJCNVT to read records from the existing BSDS, converts them to support the extended RBA and LRSN format, and writes them into the prefix.BSDS01.NEW and prefix.BSDS02.NEW data sets.
DSNTRENO
Renames the existing BSDS data sets to backup names, prefix.BSDS01.OLDFMT and prefix.BSDS02.OLDFMT.

DSNTRENN
Renames prefix.BSDS01.NEW and prefix.BSDS02.NEW to prefix.BSDS01 and prefix.BSDS02.

No job step can run unless all preceding job steps complete with return code 0.

When the installation CLIST customizes DSNTIJCB, it substitutes the BSDS names that are specified in fields COPY 1 NAME and COPY 2 NAME on installation panel DSNTIPH for prefix.BSDS01 and prefix.BSDS02. The fields allow each name to support names up to 30 bytes in length, so no special handling is needed to rename the existing data sets.

Related concepts:
- The extended 10-byte RBA and LRSN in Db2 12 (Db2 for z/OS What’s New?)

Related tasks:
- Convert BSDS records to the extended 10-byte format: DSNTIJCB

---

**Testing the data sharing group**

When you installed Db2, sample objects were created in job DSNTEJ1. The DSNTESD member of prefix.SDSNSAMP contains SQL statements that refer to these objects. Use these SQL statements to test group buffer pool caching, global lock serialization, and concurrency in the data sharing group.

Perform these tests after installing several data sharing members.

**Testing group buffer pool caching**

You can use SQL statements in member DSNTESD to verify that the group buffer pool caching operates correctly.

**Procedure**

To verify that the group buffer pool operates correctly:

1. Run SPUFI on more than one data sharing member, using member DSNTESD as the input data set. Specify AUTOCOMMIT=YES on the SPUFI panel.

Run SPUFI on the different members serially, a few seconds apart if possible. (The runs must be close enough together to avoid having Db2 close the page set because of infrequent updates. The default amount of time between updates before Db2 switches the page set from read-write to read-only is 10 minutes, as specified by the RO SWITCH TIME field of installation panel DSNTIPL1. RO SWITCH CHKPTs can also cause the data set to switch to read-only. It is a number of consecutive checkpoints.

By running the SQL statements serially, Db2 detects inter-Db2 read/write interest on the table space and index and uses the group buffer pool.

2. Verify that ITEM_COUNT increases by five after each run.

3. Issue the following command to determine whether the table space and index are using the group buffer pool:

   ```
   DIS DB(DSN8D12A) SPACENAM(DSN8S12S,XPARTS) LOCKS
   ```

---

Installation and Migration Guide
If the P-lock state is IX or SIX, then the table space and index are group buffer pool dependent, which is the correct state.

4. Issue the following command to display the statistics for GBP0:

```
DIS GBPPOOL(GBP0) GDETAIL
```

In the group detail statistics, look for non-zero values in the READS and WRITES values of the display. This indicates that Db2 is using the group buffer pool successfully for caching.

**Testing global lock serialization**

Use the SQL statements in member DSNTESD to verify that locks are acquired and released correctly across multiple data sharing members.

**Procedure**

To test global lock serialization:

1. Run SPUFI using member DSNTESD as input, on **member 1**. Specify AUTOCOMMIT=NO.
   
   Because you have inserted data into DSN8C10.PARTS but have not committed, **member 1** holds global locks.

2. Run SPUFI using member DSNTESD as input, on **member 2**. Specify AUTOCOMMIT=NO.
   
   Because **member 1** holds global locks, **member 2** must wait to perform the insert.

3. In less than one minute, commit on **member 1**. (If you wait too long to commit, **member 2** will experience a lock timeout.)
   
   The global locks should be released, and **member 2** should be able to proceed. Verify that ITEM_COUNT has increased by five.

**Testing concurrency**

You can use the SQL statements in member DSNTESD to test concurrency within the data sharing group.

**Procedure**

To test concurrency:

1. Run SPUFI concurrently on different data sharing members. Specify AUTOCOMMIT=YES.
   
   Global locking ensures that inserts to DSN8C10.PARTS are coordinated across data sharing members.

2. Verify that ITEM_COUNT increases by five each time the run completes successfully.

**Disabling and re-enabling Db2 data sharing**

Disabling Db2 data sharing is a complex procedure and is very disruptive. Do not attempt to disable Db2 data sharing without a thorough understanding of the process.
Do not make disabling Db2 data sharing part of your contingency plans for handling recovery situations. For temporary bypasses to data sharing problems, try moving to one-way data sharing, which involves stopping all but one member and having that member perform the work for the group. If one-way data sharing does not work, your data sharing problem might be resolved by performing a group restart.

The disabling procedure is included in this section for completeness, but it should rarely be used. Situations in which it might be necessary to disable Db2 data sharing are:

- You made a strategic decision to move away from a data sharing environment.
- One-way data sharing is not working.

After you disable Db2 data sharing, only one member of the data sharing group can access the previously shared data. That member is called the surviving member. Before you disable data sharing, all page sets must be converted to extended format. For details, see What to do before RBA or LRSN limits are reached (Db2 Administration Guide).

If you plan to re-enable data sharing for a group, do not change any group-wide information in the BSDS of the surviving member. Group-wide information includes the catalog alias name and the database password. It also includes the DDF name and password information, even if you do not plan to use DDF when you re-enable Db2 data sharing. If you change any of this information, you need to change the information in every member’s BSDS before you start the group.

Related tasks:

- Converting page sets to the 10-byte RBA or LRSN format (Db2 Administration Guide)
- Accessing currently committed data to avoid lock contention (Db2 Performance)

Disabling data sharing

You should only disable data sharing with caution and with a thorough understanding of the process and results. Very few situations create a need to disable Db2 data sharing, such as if you have data sharing problems and one-way data sharing does not work.

Before you begin

All page sets must be converted to extended format. For details, see What to do before RBA or LRSN limits are reached (Db2 Administration Guide).

About this task

The procedure to disable data sharing ensures that the most recent versions of all pages are externalized from the group buffer pool to disk. Db2 does not use the group buffer pool after data sharing is disabled. You must ensure that data is written to disk, or else you lose data when you start Db2 after disabling data sharing.

Important: Other members’ logs are not available to the surviving member after you disable data sharing.
You must also ensure that there is no need to recover data from information that is contained in other members' logs after you disable data sharing. To prevent the surviving member from applying inconsistent updates during recovery processing, a cold start is required to disable data sharing.

**Important:** Do not attempt to go through the installation process to re-enable data sharing after you have disabled it.

**Procedure**

To disable data sharing and return to a non-data-sharing environment:

1. Decide which member will be the surviving member of the group. The surviving member is the only member that can access previously shared data.
2. Stop all members by entering the following command for each member of the data sharing group:
   ```sql
   STOP DB2 MODE(QUIESCE)
   ```
3. Start the surviving member in maintenance mode by using the following command:
   ```sql
   START DB2 ACCESS(MAINT)
   ```
4. Make sure that data is consistent. Enter the following commands from the surviving member of the group. Do not proceed to the next step until all problems are resolved.
   - **DISPLAY GROUP**
     Ensure that the status of the surviving member is ACTIVE and that the status of all other members is QUIESCED. If the status of a non-surviving member is not QUIESCED, take the following actions, depending on the member's status:
     - If the member's status is FAILED, restart and then stop the member.
     - If the member has castout problems, indoubt units of recovery, or outstanding resynchronization problems, start that member in maintenance mode and fix the problem.
   - **DISPLAY UTILITY(*)**
     If any utility work remains for any member of the group, restart that member with ACCESS(MAINT) and either stop the utility or let it finish.
   - **DISPLAY DATABASE(*) SPACENAM(*) RESTRICT**
     If any restricted table spaces or index spaces (such as write error ranges, recovery pending status, or logical page list entries) exist, recover them from the surviving member.
5. Stop the surviving member by using the following command:
   ```sql
   STOP DB2 MODE(QUIESCE)
   ```
6. Stop any IRLMs that have not stopped by using the following command:

```plaintext
STOP irlmproc
```

7. Dismantle the data sharing group.
   a. Enter the following command to display the structures for the data sharing group:
      ```plaintext
      DXCF,STRUCTURE,STRNAME=grpname*
      ```
   b. For all structures that are still allocated (STATUS:ALLOCATED) and that still have connections (which appear as FAILED PERSISTENT), enter the following command to force the connections off of those structures:
      ```plaintext
      SETHXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
      ```
      **Important:** If your site is running z/OS with APAR OA02620 applied, you do not need to forcibly deallocate failed-persistent connections off of the lock structure. When you forcibly deallocate the lock structure (see the next step), the system deletes failed-persistent connections to the structure for you.
   c. Delete all the Db2 coupling facility structures by using the following command for each structure:
      ```plaintext
      SETHXCF FORCE,STRUCTURE,STRNAME=strname
      ```
   d. Edit the JCL in job DSNTIJGF to point to the correct BSDS data sets. DSNTIJGF is a change log inventory job that sets up the surviving member for a cold start.
      **Important:** Do not change the hex values that appear in the change log inventory CRESTART control statement. They are not real RBA values.
   e. Run job DSNTIJGF.
      After you run this job, do not try to restart any of the non-surviving members. None of those members can start successfully.

8. Change the IRLM procedure to SCOPE=LOCAL.

9. Start the surviving member with ACCESS(MAINT). Specify the old DSNZP.xxx from the non-data sharing environment, where the DSHARE subsystem parameter is set to NO. If you do not have the old DSNZP.xxx or if the surviving member is not the originating member, reassemble the surviving member's subsystem parameters with DSHARE=NO in the invocation of the DSN6GRP macro. Also, comment out all steps from the DSNTIJUZ job except for those that reassemble and link-edit the subsystem parameters.

   When you start Db2 after running DSNTIJGF, respond with Y to a cold start prompt (message DSNJ246I on the z/OS console).

   This is a cold start because Db2 increases the log RBA to a value higher than any LRSN used while sharing data. From this point on, your RBAs look like LRSNs.

10. Edit and run job DSNTIJFT, if necessary, to ensure that the surviving member's work file database is DSNDB07.

    The surviving member must use DSNDB07 as its work file database. If the work file database for the surviving member is not DSNDB07, drop that work file database and run job DSNTIJFT.
11. Verify that the surviving member works by running a subset of the Db2 12 installation verification sample jobs.

12. To establish a new recovery point, take a full or incremental image copy or non-Db2 backup of all data. Run job DSNTIJIC to image copy the Db2 catalog and directory.

**Recommendation:** Perform this step as soon as possible after data sharing is disabled.

13. Stop and restart Db2 for normal unrestricted access.

**What to do next**

After disabling Db2 data sharing, you can continue to use the group attachment name or subgroup attachment name. The group attachment name or subgroup attachment name does not need to be changed to the surviving member’s subsystem ID.

**Related concepts:**
- Data recovery after disabling Db2 data sharing

**Related tasks:**
- Verifying successful Db2 12 installation or migration with sample applications

**Data recovery after disabling Db2 data sharing**

After Db2 data sharing is disabled, you cannot recover to the current point or to a previous point in time if that recovery depends on any portion of the log that was made before you disabled data sharing.

Therefore, if the table space is updated between the time of the copy and the time you disabled data sharing, Db2 does not let you use that copy as the basis for recovery. This is why you should create a new recovery point as soon as possible after you have disabled data sharing.

**Re-enabling Db2 data sharing**

Re-enabling Db2 data sharing involves a subset of the original tasks that were required to enable data sharing. You cannot use the enabling process of installation to re-enable Db2 data sharing.

**About this task**

The following output from the original Db2 data sharing enabling procedure remains intact after disabling Db2 data sharing and does not need to be re-created or re-specified:

- Data sharing subsystem parameters (output from the CLIST processing when enabling data sharing)
- XCF definitions
- Coupling facility definitions
- RACF definitions
- Db2 catalog and directory

**Procedure**

To re-enable Db2 data sharing:
1. Edit the JCL in job DSNTIJGF.
DSNTIJGF is a change log inventory job that sets up the BSDS of the surviving member for data sharing.

2. Run job DSNTIJGF.

3. Change the IRLM procedure to SCOPE=GLOBAL.

4. Start the surviving member with the subsystem parameters that were used when data sharing was originally enabled.
   During startup, you are asked to start all other members that were not quiesced at the time that you disabled Db2 data sharing. You must start all these members.

5. Edit and run job DSNTIJFT on the surviving member to re-create the work file database for data sharing.
   See the directions in the prologue of job DSNTIJFT for information about editing this job to re-enable Db2 data sharing.
Chapter 10. Verifying successful Db2 12 installation or migration with sample applications

You can use the Db2 Db2 12 sample applications to verify the operation of Db2 after installation or new function activation. These sample applications are also called installation verification procedures (IVPs).

Before you begin

Before you can run the Db2 12 sample applications, you must activate function level 500 or higher, as described in [Migration step 29: Activate function level 500](#). Until you complete this task, continue to use sample applications such as DSNTIAUL, DSNTEP2, and DSNTEP4 from Db2 11.

About this task

Attention: Do not run the Db2 12 IVP jobs prior to activating function level V12R1M500 or V12R1M501.

Recommendations:

- During verification, run all sample applications under the same user ID. This user ID must have SYSADM authority. Otherwise, errors might occur.
- Run the IVP jobs with Db2 started in unrestricted mode because restricted mode (ACCESS(MAINT)) does not allow use of stored procedures and user-defined functions.
- If you are installing a data sharing group, run the IVP jobs after you install or migrate the originating system. You do not need to run the IVP jobs after you enable the originating system or after you install a new data sharing member.
- You must use the same compiler for each job. For example, if you use IBMCOB for DSNTEJ2C, you must use IBMCOB for all COBOL verification programs.
- After you designate a client and server subsystem, run all verification jobs on the same client and server subsystems.

Procedure

To verify a successful installation or migration:

1. For testing purposes, identify one local subsystem to act as the client and one remote subsystem to act as the server. Both subsystems must be at Db2 12. Although not recommended, the client and server subsystem can be the same if a remote subsystem is not available for testing.
2. Ensure that the client subsystem is specified in the Db2 LOCATION NAME field of installation panel DSNTIPR.
3. Ensure that the server subsystem is specified in the REMOTE LOCATION field of installation panel DSNTIPY.
4. Identify the IVP jobs that you need to run based on the attachment facilities you installed, the languages you use, and whether the sample objects exist.
5. Edit the sample jobs as necessary based on the information that is provided for the jobs.
6. Run the necessary IVP jobs.
What to do next

When you complete the verification procedure, save the verification objects. You will need them when you migrate to the next release of Db2.

Related tasks:
- Enabling Db2 data sharing
- Migration step 28: Verify migration of Db2 11 IVP objects to Db2 12 (optional)
- Installation step 25: Verify successful installation of Db2 12
- Migration step 32: Verify successful migration to Db2 12

Installation verification phases and programs

In each phase of the installation verification procedure (IVP), you must run specific programs. These programs need to be run sequentially by phase because the output of some jobs is used as input for subsequent jobs.

The IVP consist of eight phases: seven verification phases and one cleanup phase that drops sample objects (Phase 0). Each of the seven verification phases tests one or more Db2 functions or attachment facilities. Certain phases of the verification procedure might not apply to the environment in which your Db2 subsystem operates, so you might not need to complete all phases. In some cases, the steps and return codes differ when you run the fallback release and Db2 12 phases. These differences are noted under the proper phase.

Run Phase 0 (job DSNTEJ0) only if you want to remove all the verification processing that you have completed so that you can begin the verification procedure again. Phases 1-3 test the TSO and batch environments, including user-defined functions. Phase 4 is for IMS users only, and Phase 5 is for CICS users only. Phase 6 sets up the sample tables and stored procedures for distributed processing. Phase 7 tests the LOB feature with sample tables, data, and programs.

`prefix.SDSNSAMP` contains the program source.

Each phase consists of one or more batch jobs and some phases also consist of online activities. You run the same jobs regardless of whether you are installing Db2 for the first time or migrating your Db2 12 catalog to new-function mode. The installation CLIST tailored and loaded these jobs into `prefix.NEW.SDSNSAMP`, which you created during your installation or migration to new-function mode. The verification jobs use source code and data from the untailored `prefix.SDSNSAMP` library. Most Db2 sample objects have unique names to differentiate them from objects of previous releases. This allows sample programs for multiple releases to coexist. The JCL that is provided for IMS sets up transaction identifiers for the sample applications.

The following table shows the programs and jobs that are run in each phase of the verification procedure. These jobs are designed to run with minimal interaction on your part. However, before running these jobs, you should make any modifications that are suggested in the information about each job.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Job</th>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DSNTEJ0</td>
<td>DSNTIAD</td>
<td>Remove sample applications and sample schema authorizations</td>
</tr>
<tr>
<td>1</td>
<td>DSNTEJ1</td>
<td>DSNTIAD</td>
<td>Create tables</td>
</tr>
</tbody>
</table>
Table 71. Relationship of IVP phases to programs (continued)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Job</th>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DSN8CA</td>
<td>Assembler interface to call attach facility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EAE1</td>
<td>Edit exit routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8HUFF</td>
<td>Huffman compression exit routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB</td>
<td>Utilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1L</td>
<td>Dynamic SQL program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1P</td>
<td>Dynamic SQL program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1S(^1)</td>
<td>See note 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1T(^2)</td>
<td>See note 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1U</td>
<td>Create Unicode table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1U</td>
<td>Load Unicode table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ1U</td>
<td>Select Unicode table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ2A</td>
<td>Grant execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ2C</td>
<td>Generate declarations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ2D</td>
<td>Grant execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ2E</td>
<td>Prepare error message routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ2F</td>
<td>Grant execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8MDG</td>
<td>Prepare classes used by C++ phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8BECL</td>
<td>Prepare classes used by C++ phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8BE3</td>
<td>C++ phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8BF3</td>
<td>Fortran phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EDXR</td>
<td>Sample application to format and display an XML record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EDXU</td>
<td>Sample application to format and update an existing XML record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EDXI</td>
<td>Sample application to insert an XML record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EDXD</td>
<td>Sample application to delete an XML record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCLGEN</td>
<td>Generate declarations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8BP3</td>
<td>PL/I phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUAD</td>
<td>C program for ALTDATE function (current date)</td>
</tr>
</tbody>
</table>

Chapter 10. Verifying successful Db2 12 installation or migration with sample applications 735
Table 71. Relationship of IVP phases to programs (continued)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Job</th>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DSN8DUCD</td>
<td>C program for ALTDATE function (given date)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUAT</td>
<td>C program for ALTTIME function (current time).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUCT</td>
<td>C program for ALTTIME function (given time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUCY</td>
<td>C program for CURRENCY function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUTI</td>
<td>C program for TABLE_NAME, TABLE_SCHEMA, and TABLE_LOCATION functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EUDN</td>
<td>C++ program for DAYNAME function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8EUMN</td>
<td>C++ program for MONTHNAME function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEP2</td>
<td>Dynamic SQL program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUWF</td>
<td>User-defined table function sample</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8DUWC</td>
<td>C program on client for user-defined table function sample</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>SPUFI</td>
<td>Sample SPUFI input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ3C</td>
<td>Grant execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8CC</td>
<td>COBOL interface to call attachment facility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8SCM</td>
<td>COBOL connection manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8SC3</td>
<td>COBOL phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8HC3</td>
<td>COBOL organization application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ3M</td>
<td>Drop sample MQT objects, create sample MQT objects, grant authority to MQT objects, and populate sample MQT tables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNUTILB</td>
<td>Produce statistics for MQT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEP2</td>
<td>Process EXPLAINs on sample MQT tables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ3P</td>
<td>Dynamic SQL application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8SP3</td>
<td>PL/I phone application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8SPM</td>
<td>PL/I connection manager</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>DSNTEJ4C</td>
<td>Grant execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8ICx</td>
<td>Organization application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8MCx</td>
<td>Copy code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ4P</td>
<td>Grant execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8IPx</td>
<td>Organization, project applications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN8MPx</td>
<td>Copy code</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>DSNTEJ5A</td>
<td>CICS SQLCA formatter front-end</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNTEJ5C</td>
<td>Grant execution</td>
</tr>
</tbody>
</table>
Table 71. Relationship of IVP phases to programs (continued)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Job</th>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTEJ5P</td>
<td>DSN8MCx</td>
<td>DSN8MCx</td>
<td>Copy code</td>
</tr>
<tr>
<td>DSN8CPx</td>
<td>DSN8CPx</td>
<td>DSN8CPx</td>
<td>Organization, project applications</td>
</tr>
<tr>
<td>DSN8MCx</td>
<td>DSN8MCx</td>
<td>DSN8MCx</td>
<td>Copy code</td>
</tr>
<tr>
<td>6</td>
<td>DSNTEJ6</td>
<td>DSN8ED1</td>
<td>Update location column in the department table to the sample location entered at installation time</td>
</tr>
<tr>
<td>DSNTEJ6D</td>
<td>DSN8ED1</td>
<td>DSN8ED1</td>
<td>Compile, link-edit, bind, and run a sample application program that calls a stored procedure</td>
</tr>
<tr>
<td>DSNTEJ6O</td>
<td>DSN8DTS1, DSN8DTS2</td>
<td>DSN8DTS1, DSN8DTS2</td>
<td>Create and run sample for IBM Text Search for Db2 for z/OS</td>
</tr>
<tr>
<td>DSNTEJ6P</td>
<td>DSN8EPU</td>
<td>DSN8EPU</td>
<td>Compile, link-edit, bind, and run a sample application that calls DSNUTILU, the Db2 Utilities Unicode parser stored procedure</td>
</tr>
<tr>
<td>DSNTEJ6R</td>
<td>DSN8ED8</td>
<td>DSN8ED8</td>
<td>Create sample stored procedure</td>
</tr>
<tr>
<td>DSNTEJ6S</td>
<td>DSN8ED2</td>
<td>DSN8ED2</td>
<td>Register, prepare, and bind the stored procedure sample application</td>
</tr>
<tr>
<td>DSNTEJ6T</td>
<td>DSN8EPU</td>
<td>DSN8EPU</td>
<td>Compile, link-edit, bind, and run a sample application that calls DSNUTILS, a stored procedure for executing Db2 utilities</td>
</tr>
<tr>
<td>DSNTEJ6U</td>
<td>DSN8EE0</td>
<td>DSN8EE0</td>
<td>C++ class that calls DSNTIAR</td>
</tr>
<tr>
<td>DSN8EE1</td>
<td>DSN8EE1</td>
<td>DSN8EE1</td>
<td>C++ class that calls DSNUTILS</td>
</tr>
<tr>
<td>DSN8EE2</td>
<td>DSN8EE2</td>
<td>DSN8EE2</td>
<td>C++ client for DSN8EE1</td>
</tr>
<tr>
<td>DSNTEJ6W</td>
<td>DSN8ED6</td>
<td>DSN8ED6</td>
<td>Call the WLM_REFRESH stored procedure</td>
</tr>
<tr>
<td>DSNTEJ6X</td>
<td>DSN8EDA</td>
<td>DSN8EDA</td>
<td>Compile, link-edit, bind, and run a sample application that calls DSNUTILS, a stored procedure for executing Db2 utilities</td>
</tr>
<tr>
<td>DSNTEJ6Z</td>
<td>DSN8ED7</td>
<td>DSN8ED7</td>
<td>C class that calls ADMIN_INFO_SYSPARM stored procedure</td>
</tr>
<tr>
<td>DSNTEJ61</td>
<td>DSN8EC1</td>
<td>DSN8EC1</td>
<td>Create sample ODBA stored procedure</td>
</tr>
<tr>
<td>DSNTEJ62</td>
<td>DSN8EC2</td>
<td>DSN8EC2</td>
<td>Invoke sample ODBA stored procedure</td>
</tr>
<tr>
<td>DSNTEJ63</td>
<td>DSN8ES1</td>
<td>DSN8ES1</td>
<td>Run sample external SQL procedure that calculates employee earnings.</td>
</tr>
<tr>
<td>DSNTEJ64</td>
<td>DSN8ED3</td>
<td>DSN8ED3</td>
<td>Call the sample external SQL procedure, DSN8ES1 from a client</td>
</tr>
<tr>
<td>DSNTEJ65</td>
<td>DSN8ED4</td>
<td>DSN8ED4</td>
<td>Call the external SQL procedure processor, DSNTPSMP</td>
</tr>
</tbody>
</table>
Table 71. Relationship of IVP phases to programs  (continued)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Job</th>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DSN8ES2</td>
<td>DSN8ES2</td>
<td>Run external SQL procedure that calculates employee bonuses</td>
</tr>
<tr>
<td></td>
<td>DSN8ED5</td>
<td>DSN8ES2</td>
<td>Call external SQL procedure, DSN8ES2</td>
</tr>
<tr>
<td>6</td>
<td>DSNTEJ66</td>
<td>DSN8ES3</td>
<td>Create and alter a sample native SQL procedure</td>
</tr>
<tr>
<td></td>
<td>DSN8ED9</td>
<td>DSN8ES3</td>
<td>Prepare and execute a sample caller of the sample native SQL procedure</td>
</tr>
<tr>
<td>7</td>
<td>DSNTEJ7</td>
<td>DSNTIAD</td>
<td>Create sample LOB table</td>
</tr>
<tr>
<td></td>
<td>DSNTIAD</td>
<td>DSNTEJ7</td>
<td>Create synonyms, grant access to LOB tables</td>
</tr>
<tr>
<td></td>
<td>DSNUTILB</td>
<td>DSNTEJ7</td>
<td>Load sample LOB table</td>
</tr>
<tr>
<td></td>
<td>DSNUTILB</td>
<td>DSNTEJ7</td>
<td>Produce statistics for LOB table spaces</td>
</tr>
<tr>
<td></td>
<td>DSNTEJ71</td>
<td>DSNTIAD</td>
<td>Grant access to plans</td>
</tr>
<tr>
<td></td>
<td>DSN8DLPL</td>
<td>DSNTEJ71</td>
<td>Populate sample LOB table with BLOB data</td>
</tr>
<tr>
<td></td>
<td>DSN8DLTC</td>
<td>DSNTEJ71</td>
<td>Verify contents of LOB table</td>
</tr>
<tr>
<td></td>
<td>DSN8DLRV</td>
<td>DSNTEJ73</td>
<td>Grant access to plans</td>
</tr>
<tr>
<td></td>
<td>DSN8DLPV</td>
<td>DSNTEJ75</td>
<td>C employee resume application</td>
</tr>
<tr>
<td></td>
<td>DSN8CLPL</td>
<td>DSNTEJ76</td>
<td>Create COBOL LOB sample table</td>
</tr>
<tr>
<td></td>
<td>DSN8CLTC</td>
<td>DSNTEJ76</td>
<td>Create synonyms, grant access to COBOL LOB tables</td>
</tr>
<tr>
<td></td>
<td>DSN8CLRV</td>
<td>DSNTEJ77</td>
<td>Load sample COBOL LOB table</td>
</tr>
<tr>
<td></td>
<td>DSN8CLPV</td>
<td>DSNTEJ78</td>
<td>Produce statistics for COBOL BLOB table spaces</td>
</tr>
</tbody>
</table>

Notes:

1. Job DSNTEJ1S, which contains the sample JCL to run the schema processor, is not a part of the sample applications to verify installation.
2. Job DSNTEJ1T, which adds rows to SYSIBM.SYSSTRINGS for character conversion purposes, is not a part of the sample applications to verify installation.

Related concepts:

- Printing options for the sample application listings

Related tasks:

- Falling back to Db2 11
- Creating a schema by using the schema processor (Db2 Administration Guide)

Planning for verification

Before performing any of the verification phases, you must make certain decisions about your verification strategy.
About this task

Db2 system administrators and system administrators for ISPF, TSO, batch, IMS, and CICS must be involved in these decisions.

Procedure

To prepare for verification

- Determine the verification phases that you plan to perform.
  Examine the description of each verification phase in this topic, and determine which phases apply to your needs.
- Identify any phases that you want to modify before you perform them.
  Verification is designed to run with little interaction on your part. This topic does not discuss how to modify any of the phases, but you can adapt any of the seven phases to your needs. If this is your intent, identify and describe any modifications you plan to make.
- Establish additional testing steps to complete the verification.
  The verification phases and the jobs that you run to perform them are valuable tools for testing Db2. They are not a substitute for a thorough subsystem test. You must plan and perform your own additional testing to complete the verification. To help you assess which additional tests might be necessary, examine the sample applications that are provided with Db2.
- Start any Db2 databases that are not currently started.

COBOL options to use with the installation verification procedures

The Db2 COBOL samples can be run with some compiler options, which control compilation.

IBM tested these options, which are shown in the table below.

If you have a problem executing the Db2 COBOL samples, ensure that your compiler options are consistent with the Enterprise COBOL options in the table below. Remember that if you are using CICS, the options that you need to use depend on the CICS environment. To verify that you are using the correct options in your CICS environment, refer to the COBOL programming restrictions and requirements.
Table 72. Enterprise COBOL options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADV</td>
<td>NOEXIT</td>
</tr>
<tr>
<td>BUFSIZE(4096)</td>
<td>NOTYPECHK</td>
</tr>
<tr>
<td>DATA(31)</td>
<td>NOFASTSR</td>
</tr>
<tr>
<td>FLAG(I)</td>
<td>NOWORD</td>
</tr>
<tr>
<td>INTDATE(ANSI)</td>
<td>NOFLAGMIG</td>
</tr>
<tr>
<td>LANGUAGE(EN)</td>
<td>NOFLAGSTD</td>
</tr>
<tr>
<td>LINECOUNT(60)</td>
<td>NOILGEN</td>
</tr>
<tr>
<td>NOADATA</td>
<td>NUMPROC(NOPFD)</td>
</tr>
<tr>
<td>NOAWO</td>
<td>OBJECT</td>
</tr>
<tr>
<td>NOCMPR2</td>
<td>NULL</td>
</tr>
<tr>
<td>NOCOMPILE(S)</td>
<td>NOSRANGE</td>
</tr>
<tr>
<td>NOCURRENCY</td>
<td>SPACE(1)</td>
</tr>
<tr>
<td>NODBCS</td>
<td>SOURCE</td>
</tr>
<tr>
<td>NODECK</td>
<td>NOTERM1 or TERM2</td>
</tr>
<tr>
<td>NODUMP</td>
<td>TRUNC(STD)</td>
</tr>
<tr>
<td>NODYNAM</td>
<td>ZWB</td>
</tr>
</tbody>
</table>

Note:
1. Refers to jobs DSNTEJ2C, DSNTEJ3C, and DSNTEJ4C only.
2. Refers to job DSNTEJ5C only.
3. See the CICS documentation for actual options to use.

For more detailed instructions, see [Enterprise COBOL for z/OS and Language Environment Programming Guide (z/OS Language Environment Programming Guide)](رنينساتكس_(SIO _LIS_T)_(Db2 _Com_mands)

Related reference:
- [DSNH (TSO CLIST) (Db2 Commands)](رنينساتكس_(SIO _LIS_T)_(Db2 _Com_mands)

C and C++ language options to use with the installation verification procedures

The Db2 C and C++ samples can be run with some compiler options.

These options, which were tested by IBM, are shown in the following tables.

If you have a problem executing the Db2 C and C++ samples, ensure that your compiler options are consistent with the options in the following tables.

Table 73. C language options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOAGGR</td>
<td>NOEVENTS</td>
</tr>
<tr>
<td>NOALIAS</td>
<td>NOHWOPTS</td>
</tr>
<tr>
<td>ARGPARSE</td>
<td>NOMEMORY</td>
</tr>
<tr>
<td>NOCHECKOUT</td>
<td>NOPONLY</td>
</tr>
<tr>
<td>NOEXPORTALL</td>
<td>NOLINE3</td>
</tr>
<tr>
<td>NOCSECT</td>
<td>NESTINC(255)</td>
</tr>
<tr>
<td>NODECK</td>
<td>OBJECT</td>
</tr>
<tr>
<td>NODLL</td>
<td>NORENT</td>
</tr>
<tr>
<td>NOSSCOMM</td>
<td>NOSEARCH</td>
</tr>
<tr>
<td>NOUPCONV</td>
<td>NOSHOWINC</td>
</tr>
<tr>
<td></td>
<td>SOURCE</td>
</tr>
<tr>
<td></td>
<td>SPILL(128)</td>
</tr>
<tr>
<td></td>
<td>SPILL(128)</td>
</tr>
<tr>
<td></td>
<td>TERMINAL</td>
</tr>
<tr>
<td></td>
<td>NOTEST</td>
</tr>
</tbody>
</table>

740  Installation and Migration Guide
Table 73. C language options (continued)

Note:
1. This option is used by IBM C/C++ for z/OS.
2. NOPPTRACE, PPCHECK, GOTO, ACCURACY, PARM, NOENUM, NOEXTERN, TRUNC, INIT, NOPORT, GENERAL.
3. AUTO, NOREPORT, 100, 1000.
4. This option is used only by IBM AD/Cycle C/370 V1R2.
5. SYM, BLOCK, LINES, NOPATH.

The table below contains the C++ language options.

Table 74. C++ language options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARGPARSE</td>
<td>NOIDL(^1)</td>
</tr>
<tr>
<td>NOATTRIBUTE</td>
<td>NOINFO</td>
</tr>
<tr>
<td>NOCSECTS</td>
<td>NOINLRPT</td>
</tr>
<tr>
<td>EXECOPS</td>
<td>LANGLEVEL(EXTENDED)</td>
</tr>
<tr>
<td>NOEXPMPAC</td>
<td>NOLIST</td>
</tr>
<tr>
<td>NOEXPORALL</td>
<td>NLOCALE(^2)</td>
</tr>
<tr>
<td>FLAG(I)</td>
<td>LONGNAMES</td>
</tr>
<tr>
<td>NOGONUMBER</td>
<td>MARGINS</td>
</tr>
<tr>
<td>NESTINC(255)</td>
<td>OBJECT</td>
</tr>
<tr>
<td>NOSHOWINC</td>
<td>NOOFFSET</td>
</tr>
<tr>
<td>TERMINAL</td>
<td>OPTIMIZE(0)</td>
</tr>
<tr>
<td>NOID</td>
<td>SOMEINIT</td>
</tr>
<tr>
<td>NOINFO</td>
<td>SOURCE</td>
</tr>
<tr>
<td>NOINLRPT</td>
<td>NOGOSTMT</td>
</tr>
<tr>
<td>LANGLEVEL(EXTENDED)</td>
<td>NOOPTIMIZE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOSRCMSG</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOUNICLASS</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOSTORAGE</td>
</tr>
<tr>
<td>MARGINS</td>
<td>NOSYNTAX(S)</td>
</tr>
<tr>
<td>OBJECT</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOOFFSET</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOEXPMPAC</td>
<td>NONFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NONFLOW</td>
</tr>
<tr>
<td>MARGINS</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NEST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOAGGREGATE</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NONFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NLOCALE(^2)</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>LONGNAMES</td>
<td>NOFLOW</td>
</tr>
</tbody>
</table>

Note:
1. This option is used by IBM C/C++ for MVS/ESA V3R1 and subsequent releases.
2. This option is used by IBM C/C++ for z/OS and subsequent releases.

The installation CLIST customizes C++ compiler parameters in sample job DSNTEJ2E if you have specified C++ for MVS/ESA V3R2 or a subsequent release on panel DSNTP1U.

PL/I options to use with the installation verification procedures

The Db2 PL/I samples can be run with some compiler options, which control compilation.

These options, which were tested by IBM, are shown in the table below.

If you have a problem executing the Db2 PL/I samples, ensure that your compiler options are consistent with these options.

Table 75. Enterprise PL/I options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARSET(60, EBCDIC)</td>
<td>LINESIZE(55)</td>
</tr>
<tr>
<td>NOATTRIBUTES</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>NOGONUMBER</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>NOSTORAGE</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>NOSEQUENCE</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>STORAGE</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>SYNTAX(S)</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>VALUE(50656)</td>
<td>NOCOMM(55)</td>
</tr>
<tr>
<td>SOURCE</td>
<td>OBJECT</td>
</tr>
<tr>
<td>STMT</td>
<td>ODECK</td>
</tr>
<tr>
<td>TARGET(16,80)</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOAGGREGATE</td>
<td>NOFLOW</td>
</tr>
<tr>
<td>NOFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NOLIST</td>
<td>NOAGGREGATE</td>
</tr>
<tr>
<td>NONFLOW</td>
<td>NOAGGREGATE</td>
</tr>
</tbody>
</table>

Phase 0: Deleting the sample objects (DSNTEJ0)

Phase 0 consists of one job, DSNTEJ0. It frees all plans, drops all objects, and deletes data sets so that Phase 1 can be run again.
Run Phase 0 (job DSNTEJ0) only if you want to remove all the verification processing that you have done so far so that you can begin the verification procedure again. When you complete the verification procedure, save the verification objects; you need them when you migrate to the next release of Db2.

If a sample application abends while running a utility, ensure that the utility is terminated before attempting to rerun the job.

Even when DSNTEJ0 runs successfully, some of the FREE, DROP, and DELETE commands often fail because the object was not created earlier. You can ignore these errors, even though they might generate return codes of 8 or 12. Check other errors.

If DSNTEJ0 runs successfully, it produces the return codes that are shown in the table below.

Table 76. DSNTEJ0 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH00S01</td>
<td></td>
<td>0000, 0008, or 0012</td>
</tr>
<tr>
<td>PH00S02</td>
<td></td>
<td>0000, 0004, or 0008</td>
</tr>
<tr>
<td>PH00S03</td>
<td></td>
<td>0004 or 0008</td>
</tr>
<tr>
<td>PH00S04</td>
<td></td>
<td>0000, 0004, or 0008</td>
</tr>
<tr>
<td>PH00S05</td>
<td></td>
<td>0000, 0004, or 0008</td>
</tr>
<tr>
<td>PH00S06</td>
<td></td>
<td>0000, 0004, or 0008</td>
</tr>
<tr>
<td>PH00S07</td>
<td></td>
<td>0000, 0004, or 0008</td>
</tr>
<tr>
<td>PH00S08</td>
<td></td>
<td>0000, 0004, or 0008</td>
</tr>
</tbody>
</table>

If this job fails or abends, ensure that the user that is specified on the JOB statement is an authorized ID. If the name that you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTIPP1 is a primary authorization ID, use this name. If the sample authorization exit routine and RACF are installed, and if the SYSTEM ADMIN 1 and SYSTEM ADMIN 2 are known to Db2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups. Then correct any other problems, and rerun the job from the last successful step.

If the subsystem data sets were deleted before the Db2 sample objects are deleted, you must delete the data sets by using access method services commands or TSO commands. In all of the following examples, vcatalog is the catalog alias name that you specified for the CATALOG ALIAS field on installation panel DSNTIPA2. The y is either I or J.

The following access method services commands, which can be executed under TSO, delete the Db2 12 sample data sets:

DELETE 'vcatalog.DSNB00.DSNB012A.*.y0001.*'
DELETE 'vcatalog.DSNB00.DSNB011L.*.y0001.*'
DELETE 'vcatalog.DSNB00.DSNB012P.*.y0001.*'
DELETE 'vcatalog.DSNB00.DSNB012U.*.y0001.*'
DELETE 'vcatalog.DSNB04.STAFF.y0001.A001'
DELETE 'vcatalog.DSNB04.TESTSTUF.y0001.A001'
DELETE 'vcatalog.DSNB01.E.*.y0001.*'
DELETE 'vcatalog.DSNB01Y.*.y0001.*'
DELETE 'vcatalog.DSNB01X.*.y0001.*'
Phase 1: Creating and loading sample tables

Phase 1 of installation or migration verification consists of four jobs: DSNTEJ1, DSNTEJ1L, DSNTEJ1P, and DSNTEJ1U.

DSNTEJ1 invokes program DSNTIAD, which creates objects during the verification procedure. Run DSNTEJ1 before running any other sample jobs.

DSNTEJ1L and DSNTEJ1P prepare and invoke program DSNTEP2, which lists the contents of the sample tables. The difference between the jobs is that DSNTEJ1P requires the PL/I compiler and allows you to customize DSNTEP2.

DSNTEJ1U creates, populates, and tests a sample Unicode database.

Job DSNTEJ1

Job DSNTEJ1 creates all objects that are to be used by the sample verification jobs. Run this job before any other verification job.

If reason code 00C9008A continues to occur, do one of the following:
• Restart Db2 and run the job again.
• If you are using a partitioned data set (PDS) for the DSNEXIT data set, run IEBCOPY to compress the data set when no application is using the DSNEXIT data set.

The job consists of the following steps.

Table 77. Steps in job DSNTEJ1

<table>
<thead>
<tr>
<th>Step</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>Creates all objects (storage group, databases, table spaces, tables, indexes, and views) that are used by the samples.</td>
</tr>
<tr>
<td>5</td>
<td>Drops synonyms.</td>
</tr>
<tr>
<td>6</td>
<td>Creates synonyms and grants authorization on objects to PUBLIC AT ALL LOCATIONS. This step creates synonyms for the sample tables, indexes, and views, so that the currently running authorization ID can execute the sample application and grant appropriate authority. The sample dynamic SQL program DSNTIAD processes the Db2 object definitions in this step and several others.</td>
</tr>
<tr>
<td>7</td>
<td>Uses the ASMCL procedure to create DSN8EAE1, an edit exit routine.</td>
</tr>
<tr>
<td>8</td>
<td>Assembles and link-edits DSNHUFF.</td>
</tr>
<tr>
<td>9</td>
<td>Assembles and link-edits DSN8FPRC, a sample field procedure.</td>
</tr>
<tr>
<td>10</td>
<td>Prepares the sample call attachment facility assembler interface. You must link-edit ISPLINK, the ISPF interface module, with this CAF sample load module. To do this, ensure that the link-edit SYSLIB statement that retrieves the ISPF load module library in procedure DSNHASM is not commented out.</td>
</tr>
<tr>
<td>11</td>
<td>Creates the sample utility list.</td>
</tr>
<tr>
<td>12</td>
<td>Loads the programming-related tables by using the LOAD utility.</td>
</tr>
<tr>
<td>13</td>
<td>Loads the sample tables by using the LOAD utility.</td>
</tr>
</tbody>
</table>
Table 77. Steps in job DSNTEJ1 (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Checks data for referential integrity.</td>
</tr>
<tr>
<td>15</td>
<td>Establishes a quiesce point by using both log and image copies.</td>
</tr>
<tr>
<td>16</td>
<td>Makes an image copy of all the sample tables by using the COPY utility.</td>
</tr>
<tr>
<td>17</td>
<td>Establishes another quiesce point by using only image copies.</td>
</tr>
<tr>
<td>18</td>
<td>Reorganizes a table space and compiles statistics on all table spaces by using the REORG and RUNSTATS utilities.</td>
</tr>
<tr>
<td>19</td>
<td>Performs a REORG TABLESPACE with SHRLEVEL CHANGE.</td>
</tr>
<tr>
<td>20</td>
<td>Loads the sample tables by using the LOAD utility.</td>
</tr>
<tr>
<td>21</td>
<td>Sets the CURRENT RULES register and adds a check constraint using ALTER TABLE.</td>
</tr>
<tr>
<td>22</td>
<td>Checks data for referential integrity.</td>
</tr>
<tr>
<td>23</td>
<td>Checks data for check integrity.</td>
</tr>
<tr>
<td>24-27</td>
<td>Performs the operations in steps 15-18 except for the REORG on partition 3 of the Employee table space.</td>
</tr>
<tr>
<td>28</td>
<td>Unloads data from a partitioned table.</td>
</tr>
<tr>
<td>29</td>
<td>Reduces the partition key on the fourth partition of table space DSN8S81E.</td>
</tr>
<tr>
<td>30</td>
<td>Processes an online schema change to add a fifth partition to table space DSN8S81E.</td>
</tr>
<tr>
<td>31</td>
<td>Reorganizes a partitioned table space.</td>
</tr>
<tr>
<td>32</td>
<td>Performs various online schema changes, including lengthening a character field and converting an integer field to a decimal field.</td>
</tr>
<tr>
<td>33</td>
<td>Reorganizes table spaces DSN8S91R and DSN8S91S to reset advisory REORG-pending (AREO) status.</td>
</tr>
</tbody>
</table>

If DSNTEJ1 runs successfully, it produces the return codes that are shown in the following table.

Table 78. DSNTEJ1 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH01S01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S03</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S04</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S05</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S06</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S07</td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S08</td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S09</td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 78. DSNTEJ1 return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH01S10</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S11</td>
<td>IEBGENER</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S12</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S13</td>
<td>DSNUPROC</td>
<td>0004</td>
</tr>
<tr>
<td>PH01S14</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S15</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S16</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S17</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S18</td>
<td>DSNUPROC</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH01S19</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S20</td>
<td>DSNUPROC</td>
<td>0004</td>
</tr>
<tr>
<td>PH01S21</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH01S22</td>
<td>DSNUPROC</td>
<td>0004</td>
</tr>
<tr>
<td>PH01S23</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S24</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S25</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S26</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S27</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S28</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S29</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH01S30</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH01S31</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01S32</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01S33</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
</tbody>
</table>

Db2 issues the following message for every SQL statement, except for the drop synonym and insert statements:
DSNT400I SQLCODE = 0, SUCCESSFUL EXECUTION

If the synonyms in the DROP SYNONYM statements are not defined, SQL return codes of -204 result. The INSERT statements violate a check constraint on the EMP table. This results in an SQL return code of -545.

Job DSNTEJ1L

DSNTEJ1L link-edits the DSNTEP2 object deck (DSNTEP2L) to create an executable load module DSNTEP2.

DSNTEJ1L link-edits the DSNTEP4 object deck (DSNTEP4L) to create an executable load module DSNTEP4.

DSNTEJ1L also binds and runs programs DSNTEP2 and DSNTEP4. DSNTEP2 lists the sample database tables and views. DSNTEP2 is a dynamic PL/I program that
accepts SQL statements. DSNTEP2 produces a listing of the results of SELECT statements. DSNTEP4 is identical to DSNTEP2, except that it uses multi-row fetch.

Job DSNTEJ1L requires the Language Environment link-edit and run time libraries. DSNTEJ1L does not require the PL/I compiler.

If DSNTEJ1L runs successfully, it produces the return codes that are shown in the table below.

Table 79. DSNTEJ1L return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH01PS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01PS02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH01PS03</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01PS04</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

You can compare the output from this job with the sample output for DSNTEJ1L, which is found in member DSN8TJ1L in your prefix.SDSNIVPD data set.

If you run DSNTEJ1P before DSNTEJ1L, you can expect step PH01PS02 of job DSNTEJ1L to produce a return code of 0004 and the following message:

```
SQLWARNING ON GRANT COMMAND, EXECUTE FUNCTION
RESULT OF SQL STATEMENT:
DSNT404I SQLCODE = 562, WARNING: A GRANT OF A PRIVILEGE WAS IGNORED
BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR
```

If either DSNTEJ1 or DSNTEJ1L fails or abends, ensure that the user that is specified in the JOB statements is an authorized ID. If the name that you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTIP1P1 is a primary authorization ID, use this name. If the sample authorization exit routine and RACF are installed, and if the SYSTEM ADMIN 1 and SYSTEM ADMIN 2 are known to Db2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups.

Then, correct any other problems. Before rerunning DSNTEJ1, run DSNTEJ0 to drop the sample data. If you rerun DSNTEJ1L, rerun it from the last successful step.

Related reference:

DSNTEP2 and DSNTEP4 (Db2 Application programming and SQL)

Job DSNTEJ1P

If you have run DSNTEJ1L, you do not need to run DSNTEJ1P because these jobs produce the same results. The major difference is that DSNTEJ1P uses the PL/I compiler and allows you to customize DSNTEP2 and DSNTEP4.

DSNTEJ1P precompiles, compiles, and link-edits PL/I program DSNTEP2. This program then lists the sample database tables and views. It is a dynamic PL/I program that accepts SQL statements. It produces a listing of the results of SELECT statements.

If DSNTEJ1P runs successfully, it produces the return codes that are shown in the table below.
### Table 80. DSNTEJ1P return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH01PS01</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH01PS02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH01PS03</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH01PS04</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

You can compare the output from this job with the sample output for DSNTEJ1P, which is found in member DSN8TJ1P in your prefix.SDSNIVPD data set.

If you run DSNTEJ1L before DSNTEJ1P, you can expect step PH01PS02 of job DSNTEJ1P to produce a return code of 0004 and the following message:

```
SQLWARNING ON GRANT COMMAND, EXECUTE FUNCTION
RESULT OF SQL STATEMENT:
DSNT404I SQLCODE = 562, WARNING: A GRANT OF A PRIVILEGE WAS IGNORED
BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR
```

If either DSNTEJ1 or DSNTEJ1P fails or abends, ensure that the user that is specified in the JOB statements is an authorized ID. If the name that you specified for either SYSTEM ADMIN 1 or SYSTEM ADMIN 2 on installation panel DSNTPP1 is a primary authorization ID, use this name. If the sample authorization exit routine and RACF are installed, and if the SYSTEM ADMIN 1 and SYSTEM ADMIN 2 are known to Db2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups.

Then, correct any other problems. Before rerunning DSNTEJ1, run DSNTEJ0 to drop the sample data. If you rerun DSNTEJ1P, rerun it from the last successful step.

**Related reference:**

[DSNTEP2 and DSNTEP4 (Db2 Application programming and SQL)](chapter_10)

**Job DSNTEJ1U**

DSNTEJ1U creates a database, table space, and table with Unicode CCSID.

DSNTEJ1U loads data into the table from a data set that contains a full range of characters in an EBCDIC Latin-1 code page, which results in a mix of single and double-byte characters in the Unicode table. It then runs DSNTEP2 to select and display the data in hex format.
If DSNTEJ1U runs successfully, it produces the return codes that are shown in the table below.

Table 81. DSNTEJ1U return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH01US01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01US02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH01US03</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH01US04</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Phase 2: Testing the batch environment

Phase 2 of the installation or migration verification process consists of several jobs. Run the jobs to test the program preparation procedures for various languages.

If any of the Phase 2 jobs fail or abend, be sure that the user that is specified in the JOB statements is authorized. Use the name that you specified for either the SYSTEM ADMIN 1 option or the SYSTEM ADMIN 2 option on installation panel DSNTIPP1. Then correct any other problems, and rerun the jobs from the last successful step.

Job DSNTEJ2A

DSNTEJ2A tests the assembler program preparation procedures.

This job prepares and invokes program DSNTIAUL, which demonstrates the use of dynamic SQL in assembler to unload the data from tables or views. It also generates LOAD utility statements so the data can be loaded into another table. DSNTEJ2A then uses the LOAD utility to put data into copies of the unloaded tables.

If DSNTEJ2C runs successfully, it produces the return codes that are shown in the table below.

Table 82. DSNTEJ2A return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREPUNL</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>BINDUNL</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>DELETE</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>CREATE</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>UNLOAD</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>EDIT</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>LOAD</td>
<td>DSNUPROC</td>
<td>0004</td>
</tr>
</tbody>
</table>

You can compare the output from this job with the sample output for DSNTEJ2A found in member DSN8TJ2A in your prefix.SDSNIVPD data set.

Related reference:
Job DSNTEJ2C

Job DSNTEJ2C tests the COBOL program preparation procedures. This job runs the phone application.

The phone application processes a table of telephone numbers, executing various types of SELECT statements and producing the corresponding listings. It can also update a phone number.

If DSNTEJ2C runs successfully, it produces the return codes that are shown in the following table.

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02CS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02CS02</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02CS03</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02CS04</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.

You can compare the output from this job with the sample output for DSNTEJ2C found in member DSN8TJ2C in your prefix.SDSNIVPD data set.

Related concepts:
COBOL options to use with the installation verification procedures

Related tasks:
Working with the phone application scenario

Job DSNTEJ2D

Job DSNTEJ2D tests the C program preparation procedures.

You must have sequence numbering on to run this job from an ISPF session. The C job runs only the phone application.

If DSNTEJ2D runs successfully, it produces the return codes that are shown in the table below.
Table 84. DSNTE2D return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02DS01</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH02DS02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH02DS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

You can compare the output from this job with the sample output for DSNTEJ2D found in member DSN8TJ2D in your prefix.SDSNIVPD data set.

**Related concepts:**

Job DSNTEJ2E

Job DSNTEJ2E tests the C++ program preparation procedures.

You must have sequence numbering on to run this job from an ISPF session. The C++ job runs only the phone application.

If DSNTEJ2E runs successfully, it produces the return codes that are shown in the table below.

Table 85. DSNTE2E return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02ES01</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH02ES02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH02ES03</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02ES04</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 85. DSNTE2E return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
</table>

Notes:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.
2. The PLKED step does not exist in the job if the value of the CPP LE PRELINKER PHASE field is set to NO.

You can compare the output from this job with the sample output for DSNTEJ2E found in member DSN8TJ2E in your prefix.SDSNIPD data set.

Related concepts:
- Job DSNTEJ2C

Job DSNTEJ2F

Job DSNTEJ2F tests the Fortran program preparation procedures.

The FORTRAN job runs only the phone application.

If DSNTEJ2F runs successfully, it produces the return codes that are shown in the table below.

Table 86. DSNTE2F return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02FS01</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH02FS02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>FORT</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02FS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

You can compare the output from this job with the sample output for DSNTEJ2F found in member DSN8TJ2F in your prefix.SDSNIPD data set.

Related concepts:
- Job DSNTEJ2C

Job DSNTEJ2H

Job DSNTEJ2H tests the XML program preparation procedures.

If this job runs successfully, it produces the following return codes:

Table 87. DSNTEJ2H return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02HS01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>
### Table 87. DSNTEJ2H return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02HS02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS03</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS04</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS05</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS06</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS07</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS08</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02HS09</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

---

### Job DSNTEJ2P

Job DSNTEJ2P tests the PL/I program preparation procedures.

The PL/I job runs the phone application.

If DSNTEJ2P runs successfully, it produces the return codes that are shown in the table below.

### Table 88. DSNTEJ2P return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02PS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02PS02</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH02PS03</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02PS04</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02PS05</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the PLI LE PRELINKER PHASE field is set to NO.
You can compare the output from this job with the sample output for DSNTEJ2P found in member DSN8TJ2P in your prefix.SDSNIVPD data set.

**Job DSNTEJ2U**

DSNTEJ2U prepares and tests several sample user-defined functions and a driver program that exercises the user-defined functions.

In order for the installation CLIST to generate job DSNTEJ2U, you must complete the following steps during installation:

- Specify that the host has access to C/C++ for z/OS on installation panel DSNTIPU.
- Specify the name of the default WLM environment on installation panel DSNTIPX.

The sample user-defined functions are:

**Function**

**Description**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTDATE</td>
<td>Returns the current date in a user-specified format or converts a user-specified date from one format to another.</td>
</tr>
<tr>
<td>ALTTIME</td>
<td>Returns the current time in a user-specified format or converts a user-specified time from one format to another.</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>Formats a floating point number as a currency value.</td>
</tr>
<tr>
<td>DAYNAME</td>
<td>Returns the day of the week for a user-specified date in ISO format.</td>
</tr>
<tr>
<td>MONTHNAME</td>
<td>Returns the month for a user-specified date in ISO format.</td>
</tr>
<tr>
<td>TABLE_LOCATION</td>
<td>Returns the location name of a table, view, or undefined object found after resolving aliases for a user-specified object.</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>Returns the name of a table, view, or undefined object found after resolving aliases for a user-specified object.</td>
</tr>
<tr>
<td>TABLE_SCHEMA</td>
<td>Returns the schema name of a table, view, or undefined object found after resolving aliases for a user-specified object.</td>
</tr>
<tr>
<td>WEATHER</td>
<td>Returns sample weather data obtained from a TSO data set by way of demonstrating the usefulness of a user-defined function table function.</td>
</tr>
</tbody>
</table>

If you do not have C++ installed, skip steps PH02US08 and PH02US09. Also remove all statements that refer to DAYNAME and MONTHNAME from part DSNTESU in the prefix.SDSNSAMP library.

Job DSNTEJ2U consists of the steps that are listed in the table below.
### Table 89. Steps in job DSNTEJ2U

<table>
<thead>
<tr>
<th>Step</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drops all specific sample user-defined functions.</td>
</tr>
<tr>
<td>2</td>
<td>Creates and registers all sample user-defined functions. Grants EXECUTE authority to PUBLIC for the sample user-defined functions.</td>
</tr>
<tr>
<td>3-10</td>
<td>Prepares the eight external programs used by specific user-defined functions.</td>
</tr>
<tr>
<td>11</td>
<td>Binds the package for the TABLE_NAME, TABLE_SCHEMA, and TABLE_LOCATION functions. These are the only sample functions that issue SQL statements. This step also grants EXECUTE authority on these two samples to PUBLIC.</td>
</tr>
<tr>
<td>12</td>
<td>Invokes DSNTEP2 to exercise the sample user-defined functions.</td>
</tr>
<tr>
<td>13</td>
<td>Prepares DSN8DUWF, the external module for the sample user-defined table function, WEATHER.</td>
</tr>
<tr>
<td>14</td>
<td>Prepares DSN8DUWC, a sample client function for statically invoking the WEATHER user-defined table function.</td>
</tr>
<tr>
<td>15</td>
<td>Binds the package and plan for DSN8DUWC and grants the necessary authorities.</td>
</tr>
<tr>
<td>16</td>
<td>Invokes DSN8DUWC.</td>
</tr>
</tbody>
</table>

If DSNTEJ2U runs successfully, it produces the return codes that are shown in the table below.

### Table 90. DSNTEJ2U return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02US01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02US02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH02US03 - PH02US07</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02US08 - PH02US09</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02US10</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02US11</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH02US12</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH02US13</td>
<td></td>
<td>0004</td>
</tr>
</tbody>
</table>

754  
Installation and Migration Guide
Table 90. DSNTEJ2U return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH02US14</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH02US15</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH02US16</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Notes:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.
2. The PLKED step does not exist in the job if the value of the CPP LE PRELINKER PHASE field is set to NO.

You can compare the output from this job with the sample output for DSNTEJ2U found in member DSN8TJ2U in your prefix.SDSNIVPD data set.

Related concepts:
- Sample user-defined functions (Db2 SQL)
- Steps to creating and using a user-defined function (Db2 Application programming and SQL)

Related reference:
- DSNTIPU: Host language data set names panel 1
- DSNTIPX: Routine parameters panel

Phase 3: Testing SPUFI, DRDA access, dynamic SQL, and TSO

Phase 3 of the installation or migration verification process allows you to test SPUFI (SQL Processor Using File Input) and DRDA access. You can also run dynamic SQL statements, run the phone application in TSO, and bind packages at the local and remote locations.

SPUFI is a facility of Db2I. You can only run SPUFI under ISPF. You can run dynamic SQL whether or not you have ISPF.

Related tasks:
- Testing SPUFI

Testing SPUFI

During Phase 3 of the installation or migration verification process, you can test SPUFI to ensure that it is working properly.

Procedure

To test SPUFI:
1. Log on to TSO.
2. Enter ISPF (this might be done for you, depending on your site's standard practice).
3. On the DB2I defaults panel, change the Db2 name to the Db2 subsystem name you entered on panel DSNTIPM during installation. Then select DB2I on the ISPF Primary Option Menu.

4. Select SPUFI on the DB2I menu.

5. Enter the library name 'prefix.NEW.SDSNSAMP(DSNTESA)' as input to SPUFI on line 1, the DATASET NAME parameter. If your site uses the comma as a decimal point, the library name entered must be for the tailored version of job DSNTESA that was modified by the installation CLIST.

6. Define an output data set name on line 4, the output DATASET NAME parameter of the panel. This allows you to review the output.

7. Press ENTER, and examine the results. These SQL statements require a significant amount of Db2 processing; you could have to wait for the output.

8. Run steps 5, 6, and 7 three times:
   - Once with member DSNTESA, which uses a set of SQL statements to create a short-lived table space and table.
   - Once with member DSNTESC, which creates objects that support EXPLAIN information. If you are migrating from Db2 11, DSNTESC is customized by the installation CLIST to migrate your Db2 11 PLAN_TABLE, DSN_FUNCTION_TABLE, DSN_STATEMENT_TABLE, and DSN_STATEMENT_CACHE_TABLE to Db2 12 using a schema name of DSN8C10. If these Db2 11 tables have a different schema name, edit DSNTESC to use your schema name. Migration of these tables from a prior release should not be performed more than once. Also, some table spaces that are created by DSNTESC specify non-default buffer pools, BP8K1 and BP16K1. Verify that these buffer pools are enabled on Db2 before you process this DDL.
   - Once with member DSNTESE, which retrieves the EXPLAIN information.

What to do next

If any step fails or abends, be sure that the Db2 subsystem name is specified in the NAME field on the DB2I Defaults panel.

If you must drop either a Db2 11 or Db2 12 PLAN_TABLE, remove the appropriate comments from the job to issue the DROP statements.

Also, make sure that the user ID that you are using is authorized. If the name that you specified for either the SYSADM or SYSADM2 subsystem parameter is a primary authorization ID, use this name. If the sample authorization exit and RACF are installed, and both SYSADM and SYSADM2 are known to Db2 as secondary authorization IDs, you can run these jobs under a user ID in either of these RACF groups. Then, correct any other problems and rerun the scenario from the last successful step.

Related concepts:
Dynamic SQL statements: DSNTESA, DSNTESQ

Related tasks:
Creating EXPLAIN tables (Db2 Performance)

Related reference:
EXPLAIN tables (Db2 Performance)
Dynamic SQL and the ISPF/CAF application

The Phase 3 jobs of the installation or migration verification process install the ISPF/CAF sample application. This sample consists of an assembler or COBOL call attachment facility (CAF) interface, a connection manager program, and the phone application. The sample also includes the distributed application that uses DRDA access.

Job DSNTEJ1 prepares the assembler interface, and job DSNTEJ3C prepares the COBOL interface. The connection manager program and the phone application each exist in COBOL and PL/I. Job DSNTEJ3C prepares the COBOL version; job DSNTEJ3P prepares the PL/I version. The distributed application using DRDA access is written in COBOL.

Related concepts:
COBOL options to use with the installation verification procedures

Jobs DSNTEJ3C and DSNTEJ3P

Job DSNTEJ3C prepares a COBOL ISPF/CAF sample application and job DSNTEJ3P prepares a PL/I ISPF/CAF sample application.

To prepare for the distributed sample application, job DSNTEJ3C binds a package at the local (client) and remote (server) subsystems. For concurrent installations at two Db2 locations, designate one location as the client and the other location as the server. Both the client and server subsystems must be running Db2 12. Because DSNTEJ3C does a remote bind, you must set up your client and server systems for remote communication before running this job. The sample jobs DSNTEJ1 and DSNTEJ2C or DSNTEJ2P must have been run on the server subsystem.

PH03CS06, the last job step of DSNTEJ3C, binds the sample organization application at a server Db2 subsystem if the following conditions are met. If these conditions are not met, job step PH03CS06 is removed by the installation CLIST.

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the REMOTE LOCATION field of panel DSNTIPY and it does not match the name that is specified in the Db2 LOCATION NAME field of panel DSNTIPR.

If DSNTEJ3C runs successfully, it produces the return codes that are shown in the following table.

Table 91. DSNTEJ3C return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH03CS01</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH03CS02</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 91. DSNTEJ3C return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH03CS03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>COB</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PLKED</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>LKED</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH03CS04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>COB</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PLKED</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>LKED</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH03CS05</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH03CS06</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.

Step PH03CS06 can give a return code of 0004 if sample job DSNTEJ1 was not run on the remote system. For testing, you should run job DSNTEJ1 on the remote system.

If job DSNTEJ3P runs successfully, it produces the return codes that are shown in the following table.

Table 92. DSNTEJ3P return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH03PS01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPLI</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PLI</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PLKED</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>LKED</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH03PS02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPLI</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PLI</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PLKED</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>LKED</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH03PS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

Related tasks:
Connecting distributed database systems

Related reference:
DSNTIPY: Job editing panel
Job DSNTEJ3M

Job DSNTEJ3M creates, populates, and processes a database that demonstrates the use of Db2 materialized query tables (MQTs).

Before you run job DSNTEJ3M, you must create a PLAN_TABLE. To create this table, specify the member DSNTESC as an input data set name to SPUFI.

If DSNTEJ3M runs successfully, it produces the return codes that are shown in the table below.

<table>
<thead>
<tr>
<th>Table 93. DSNTEJ3M return codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step</strong></td>
</tr>
<tr>
<td>PH03MS00</td>
</tr>
<tr>
<td>PH03MS01</td>
</tr>
<tr>
<td>PH03MS02</td>
</tr>
<tr>
<td>PH03MS03</td>
</tr>
<tr>
<td>PH03MS04</td>
</tr>
<tr>
<td>PH03MS05</td>
</tr>
<tr>
<td>PH03MS06</td>
</tr>
<tr>
<td>PH03MS07</td>
</tr>
</tbody>
</table>

Starting an application in an ISPF/TSO environment

You must have access to ISPF load module libraries to run the ISPF/CAF sample application.

Procedure

To start an application in a TSO environment:

1. Enter a CALL command for option 6 of the ISPF primary option menu.
2. To start the COBOL phone sample version of the connection manager, enter the following command:
   
   ```
   CALL 'prefix.RUNLIB.LOAD(DSN8SCM)'
   ```

3. To start the PL/I phone sample version of the connection manager, enter the following command:
   
   ```
   CALL 'prefix.RUNLIB.LOAD(DSN8SPM)'
   ```

   After you enter one of these commands, Db2 displays the sample applications panel, which is shown in the following figure.
Choosing option 1 or 2 on the sample applications panel during Phase 3 invokes either the COBOL or the PL/I version of the phone application.

Choosing option 3 on the sample applications panel during Phase 6 invokes the COBOL organization application, which uses DRDA access to distributed data.

Choosing options 4, 5, 6, and 7 on the sample applications panel during Phase 7 invokes the "Employee Resume" and "Employee Photo" applications, which processes LOB data. Options 4 and 5 access the C language sample applications, and options 6 and 7 access the COBOL language sample applications. You must run job DSNTEJ73 before you can access option 4. You must run job DSNTEJ75 before you can access option 5. You must run job DSNTEJ77 before you can access option 6. You must run job DSNTEJ78 before you can access option 7.

Related tasks:
- Working with the distributed organization application scenario
- Working with the employee resume and photo scenarios
- Make DB2I panels, messages, and load modules available to ISPF and TSO
- Working with the phone application scenario

Phase 4: Testing the IMS environment

Phase 4 of the installation or migration verification process installs the sample IMS transactions for both COBOL and PL/I.

In the PL/I version, the phone application discussed in Phase 2 is also installed as an online transaction.

Related tasks:
- Working with the phone application scenario

Jobs DSNTEJ4C and DSNTEJ4P

Job DSNTEJ4C is for COBOL; DSNTEJ4P is for PL/I. Both jobs complete the same functions.

Jobs DSNTEJ4C and DSNTEJ4P complete the following tasks:
- Precompile, compile, and link-edit the IMS online applications.
- Bind the IMS online applications.
- Create the message format service (MFS) panels for the online applications.
• Run the required PSBGEN and ACBGEN.

Select the proper job and define the applications and transactions to IMS. Member DSN8FIMS in prefix.SDNSAMP contains information to assist in the definition step.

The verification transactions are single mode, single segment, and nonconversational.

**Recommendation:** Use SSM error option R because the program handles any errors. A resource translation table is not required.

Invoke the transaction by using the FORMAT command. The programs accept several lines of input on the first panel and display the results after you press ENTER.

If DSNTEJ4C runs successfully, it produces the return codes that are shown in the following table.

**Table 94. DSNTEJ4C return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH04CS01</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04CS02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04CS03</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04CS04</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH04CS05</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH04CS06</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0004</td>
</tr>
<tr>
<td>PH04CS07</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0004</td>
</tr>
<tr>
<td>PH04CS08</td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0000</td>
</tr>
<tr>
<td>PH04CS09</td>
<td>G</td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.
For DSNTIJ4C, the warning code that is expected from the precompiler step PH04CS01 is:

```
DB2 SQL PRECOMPLIER MESSAGES
DSNH0531 W NO SQL STATEMENTS WEREN T FOUND
```

If DSNTIJ4P runs successfully, it produces the return codes that are shown in the following table.

**Table 95. DSNTIJ4P return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH04PS01</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS02</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS03</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS04</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS05</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS06</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS07</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS08</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH04PS09</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 95. DSNTEJ4P return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH04PS10</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS11</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0004</td>
</tr>
<tr>
<td>PH04PS12</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH04PS13</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH04PS14</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH04PS15</td>
<td>S1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>S2</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH04PS16</td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0000</td>
</tr>
<tr>
<td>PH04PS17</td>
<td>G</td>
<td>0000</td>
</tr>
<tr>
<td>PH04PS18</td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0000</td>
</tr>
<tr>
<td>PH04PS19</td>
<td>G</td>
<td>0000</td>
</tr>
<tr>
<td>PH04PS20</td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>0000</td>
</tr>
<tr>
<td>PH04PS21</td>
<td>G</td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

For DSNTEJ4P, the warning code expected from the precompiler step PH04PS01 is:
DB2 SQL PRECOMPIlER MESSAGES
DSNH0531 W NO SQL STATEMENTS WERE FOUND

If either job DSNTEJ4C or job DSNTEJ4P fails or abends, rerun the jobs from the last successful step.

Related concepts:
- COBOL options to use with the installation verification procedures

Starting an application in an IMS environment

After logging on to IMS, you can start the organization application or project application.
Procedure

To start an application in an IMS environment:

Enter an IMS FORMAT command

The FORMAT commands are:

- `/FORMAT DSN8IPGO`, which starts the PL/I organization version
- `/FORMAT DSN8ICGO`, which starts the COBOL organization version.

When you enter either of these two commands, the panel that is shown in the following figure is displayed.

When the following command is entered, the panel that is shown in the figure below is displayed.

```
/FORMAT DSN8IPFO
```

starts the PL/I projects version.

**Using the phone application in IMS**

When you use IMS, information is interactively processed.

Procedure

To begin using the phone application in IMS:

Clear the screen and type in a FORMAT command. The FORMAT command that starts PL/I phone application is:

```
/FORMAT DSN8IPNO
```

When the FORMAT command is entered, the panel that is shown in the following figure is displayed.
Phase 5: Testing the CICS environment

Phase 5 of the installation or migration verification process tests the CICS environment. Phase 5 installs the sample applications for COBOL and PL/I, and it prepares the CICS SQLCA formatter front-end.

Job DSNTEJ5A

Job DSNTEJ5A assembles and link-edits DSNTIAC, the CICS SQLCA formatter front-end. DSNTEJ5A also assembles and link-edits the RCT and optionally adds the sample definitions to the CSD.

Use DSNTIAC as an alternative to DSNTIAR when you want CICS services to do storage handling and program loading. If you are using CICS Version 4 or CICS Transaction Server 1.1, you need to modify job DSNTEJ5A to use steps DSN8FRCT and DSN8FRDO. You might need to tailor step DSN8FRDO for your system.

If DSNTEJ5A runs successfully, it produces the return codes that are shown in the table below.

Table 96. DSNTEJ5A return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH05AS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05AS02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05AS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Jobs DSNTEJ5C and DSNTEJ5P

Job DSNTEJ5C installs the sample application transactions in COBOL and prepares the organization application. Job DSNTEJ5P installs the transactions in PL/I and prepares the organization, project, and phone applications.

Both phase 5 jobs perform the following functions:
- Compile and link-edit the CICS online applications
- Bind the CICS online applications
- Create the BMS maps for the online applications.

Select the proper job, and define transactions, programs, and BMS maps to CICS.
prefix=DSNSAMP members DSN8FPPT, DSN8FPCT, and DSN8FRCT contain the respective PPT, PCT, and RCT entries required for the phase 5 applications. These members help you perform the definition step. Make sure that the subsystem ID (SUBID) in the RCT entry matches your Db2 subsystem ID.

If DSNTMEJ5C runs successfully, it produces the return codes that are shown in the following table.

*Table 97. DSNTMEJ5C return codes*

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPG</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>MAPD</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>DSNH</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>BIND</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>MAPGP</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>MAPGL</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>MAPDP</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>MAPDL</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

If DSNTMEJ5C fails or abends, rerun the job from the last successful step. To receive more prepare-time detail from DSNTMEJ5C, change the parameters TERM(LEAVE) and PRINT(LEAVE) to TERM(TERM) and PRINT(TERM).

If DSNTMEJ5P runs successfully, it produces the return codes that are shown in the following table.

*Table 98. DSNTMEJ5P return codes*

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH05PS01</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS02</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS03</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS04</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS05</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS06</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS07</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS08</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS09</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS10</td>
<td></td>
<td>0004</td>
</tr>
</tbody>
</table>
Table 98. DSNTEJ5P return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH05PS11</td>
<td>PPLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS12</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS13</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS14</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS15</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS16</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS17</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS18</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS19</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS20</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS21</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH05PS22</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH05PS23</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS24</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PH25</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS26</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PH27</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS28</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PH29</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS30</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PH31</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS32</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 98. DSNTEJ5P return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH05PH33</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS34</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PH35</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH05PS36</td>
<td>ASSEM</td>
<td>0000</td>
</tr>
<tr>
<td>PH05PH37</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

If DSNTEJ5P fails or abends, rerun the job from the last successful step. You might find it convenient to break up DSNTEJ5P and run only the unsuccessful steps.

Related concepts:
COBOL options to use with the installation verification procedures

Related reference:

DSNH (TSO CLIST) (Db2 Commands)

Starting an application in a CICS environment

After logging on to CICS, you can start an organization application or project application in the CICS environment.

Procedure

To start an application in a CICS environment:

Enter a CICS transaction code.
The CICS transaction codes are:
- D8PP, which starts the PL/I project version
- D8PS, which starts the PL/I organization version
- D8CS, which starts the COBOL organization version.

When these transaction codes are entered, the panels that are shown in the following two figures are displayed.
Using the phone application in CICS
When you use CICS, information is interactively processed.

Procedure
To begin using the phone application in CICS:

Clear the screen and type in the transaction code:
D8PT

You can change the transaction codes when you install Db2. Check with your system administrator to find out if they have been changed from those shown.

Using CICS storage-handling facilities
You can use the CICS storage-handling facilities when you run the CICS sample applications.

Procedure
To use the CICS storage-handling facilities when running the CICS sample applications:
1. Change your DSNTIAR calls to DSNTIAC calls in DSN8MCxx and DSN8MPxx. The calls should look like this:
   CALL DSNTIAC(EIB,COMAREA,SQLCA,MSG,LRECL)
2. Define DSNTIAC and DSNTIA1 in the CSD.
3. Rerun job DSNTEJ5C or job DSNTEJ5P.

Phase 6: Accessing data at a remote site
You can use phase 6 of the Db2 installation or migration verification process to verify that the features of DRDA access are working correctly, to prepare and call sample stored procedures, and to prepare and execute sample callers of certain Db2-supplied stored procedures.

During this optional phase, you test accessing data at a remote site by using the DRDA access application (job DSNTEJ6 in conjunction with job DSNTEJ3C).
You can use the following jobs to test creating and calling stored procedures at either the local site or a remote site.

**DSNTEJ6S and DSNTEJ6P**
Tests the stored procedure without result set sample

**DSNTEJ6T and DSNTEJ6D**
Tests the stored procedure with result set sample

**DSNTEJ61 and DSNTEJ62**
Tests the stored procedure for IMS Open Database Access

**DSNTEJ63 and DSNTEJ64**
Tests the external SQL procedure batch sample

**DSNTEJ65**
Tests the external SQL procedures processor invocation sample

**DSNTEJ66**
Tests the stored procedure that prepares a sample native SQL procedure and caller

You can use the following jobs to test calling Db2-supplied stored procedures at the local site:

**DSNTEJ6U, DSNTEJ6R, DSNTEJ6V, and DSNTEJ6X**
Tests the stored procedure for invoking utilities

**DSNTEJ6W**
Tests the stored procedure for invoking WLM_REFRESH

**DSNTEJ6Z**
Tests the stored procedure for formatting the IFCID 106 trace record to obtain current Db2 configuration settings

### ISPF panel requirements

The installation CLIST prepares samples DSNTEJ6, DSNTEJ6S, DSNTEJ6P, DSNTEJ6T, DSNTEJ6D, DSNTEJ6Z, DSNTEJ61, DSNTEJ62, DSNTEJ63, DSNTEJ64, DSNTEJ65 and DSNTEJ66 only if you specify COMMAND or AUTO in the DDF STARTUP OPTION field of panel DSNTIPR. If you specify NO, the installation CLIST does not prepare these samples. In this case, you should not try to run these samples.

The installation CLIST prepares the stored procedures samples only if you specify a sample WLM environment name in the WLM ENVIRONMENT field on panel DSNTIPX. If you replace the default value with blanks in the WLM ENVIRONMENT field, you cannot start the sample WLM-established stored procedures address space until you update the subsystem parameter.

The installation CLIST tailors the phase 6 sample jobs according to the information that you specify in the REMOTE LOCATION field of panel DSNTIPY. The guidelines for this field are:

- If the field is blank, the installation CLIST only customizes phase 6 sample jobs DSNTEJ63, DSNTEJ64, DSNTEJ65, DSNTEJ66, DSNTEJ6R, DSNTEJ6U, DSNTEJ6V, DSNTEJ6W, and DSNTEJ6Z. Jobs DSNTEJ63, DSNTEJ64, DSNTEJ65, and DSNTEJ66 use the local location name when the REMOTE LOCATION field is blank. Jobs DSNTEJ6R, DSNTEJ6U, DSNTEJ6V, DSNTEJ6W, and DSNTEJ6Z do not use a remote location name.
If the value in the field is the same as the location name for the Db2 subsystem that you are installing (DB2 LOCATION NAME field of panel DSNTIPR), the stored procedures samples are prepared and customized for local use. However, the DRDA access sample is not prepared, including job DSNTEJ6 and the DRDA access component of job DSNTEJ3C.

If the value in the field is different from the DB2 LOCATION NAME field, the installation CLIST prepares the phase 6 samples assuming that the remote location is the server and that the local system is the client.

The following table summarizes the ISPF panel values that are required for the installation CLIST to prepare the phase 6 sample jobs.

Table 99. IVP phase 6 job requirements. The installation CLIST only customizes the phase 6 sample jobs if certain fields on the ISPF panels meet requirements.

<table>
<thead>
<tr>
<th>Job name</th>
<th>Requires COMMAND or AUTO in the DDF STARTUP OPTION field of panel DSNTIPR?</th>
<th>Requires a name in the REMOTE LOCATION field of panel DSNTIPY?</th>
<th>Requires a name in the WLM ENVIRONMENT field of panel DSNTIPX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTEJ6</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNTEJ6S</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ6P</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ6T</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ6D</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ6R</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ6U</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNTEJ6V</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNTEJ6W</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ6Z</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ61</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ62</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ63</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ64</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ65</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DSNTEJ66</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Testing multiple subsystems

If you are installing and testing two Db2 subsystems concurrently, you must designate one as the server and the other as the client. If you change these designations during your testing, your results will be unpredictable. Verify that your VTAM APPL statement has the parameter SYNCLVL=SYNCPT defined. This setting allows updates at several locations.

DRDA access sample

The distributed sample application that uses DRDA access is executed as part of Phase 6 of the installation or migration verification process.
The application is prepared in Phase 3 as part of DSNTIJ3C. Before this application can be run correctly as a DRDA access sample, you must run job DSNTIJ6 at both the local and remote sites to tailor the DEPT sample table for use in a distributed environment.

To set up your samples testing for concurrent installations at two Db2 locations, follow these guidelines:

- Designate one location as the requester (the client) and the other location as the server.
- Run the client version of DSNTIJ6 at the client site only; do not run the client version of DSNTIJ6 at the server.
- Edit the server version of DSNTIJ6 at the remote server site; do not run the server version of DSNTIJ6 at the client.

Locate the following text in the server version of DSNTIJ6 within step PH06S01:

```
UPDATE DEPT SET LOCATION = (your remote location name) WHERE DEPTNO = 'F22';
UPDATE DEPT SET LOCATION = (your location name) WHERE LOCATION = '';
```

This text should be replaced with:

```
UPDATE DEPT SET LOCATION = (your location name) WHERE DEPTNO = 'F22';
UPDATE DEPT SET LOCATION = (your remote location name) WHERE LOCATION = '';
```

**Job DSNTIJ6**

Job DSNTIJ6 consists of a single step that updates the location column in the sample DEPT table at the client and server Db2 subsystems.

The employees in department F22 are recorded as working at the server location that was entered in the REMOTE LOCATION field of panel DSNTIPY. All other employees are recorded as working at the client location.

The installation CLIST edits this job only if the following conditions are met:

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the REMOTE LOCATION field of panel DSNTIPY and it does not match the name specified in the DB2 LOCATION NAME field of panel DSNTIPR.

Before you run DSNTIJ6, ensure that the server Db2 subsystem’s DSNTIJ1 job has been run on the server Db2 subsystem. Run the client Db2 subsystem’s DSNTIJ6 job only on the server Db2 subsystem.

If DSNTIJ6 runs successfully, it produces the return code that is shown in the table below.

**Table 100. DSNTIJ6 return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06S01</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>
Related tasks:
Starting an application in an ISPF/TSO environment in phase 6

Related reference:
DSNTIPR: Distributed data facility panel
DSNTIYY: Job editing panel

Starting an application in an ISPF/TSO environment in phase 6
You must have access to ISPF load module libraries in order to run the ISPF/CAF sample application.

Procedure

To start the sample application:
1. Enter a CALL command from option 6 of the ISPF primary option menu.
2. To start the COBOL sample version of the connection manager, enter the following:
   
   ```
   CALL 'prefix.RUNLIB.LOAD(DSN8SCM)'
   ```
   
   After you enter this command, Db2 displays the sample applications panel. Choosing option 3 on the sample applications panel during Phase 6 invokes the COBOL organization application, which uses DRDA access for distributed data.

Related tasks:
Working with the distributed organization application scenario
Starting an application in an ISPF/TSO environment in phase 6
Make DB2I panels, messages, and load modules available to ISPF and TSO

Stored procedure samples

Db2 provides a set of stored procedure sample applications. These sample applications demonstrate different ways that a stored procedure can be used by a client to issue Db2 commands to a Db2 server.

Most of these applications prepare and run two programs. One program prepares a stored procedure, and one program executes a client program that calls the stored procedure and returns some response.

Db2 provides the following stored procedure samples:

- A sample stored procedure without a result set
- A sample stored procedure with a result set
- Samples that invoke the following utilities stored procedures:
  - DSNUTILS
  - DSNUTILU
  - DSNUTILV
- A sample that invokes the stored procedure WLM_REFRESH, which refreshes the WLM environment
- A sample that invokes the stored procedure ADMIN_INFO_SYSPARM, which lists the settings of subsystem parameters
- A sample stored procedure that uses IMS Open Database Access (ODBA) support
- Two samples of external SQL procedures
All of these samples are part of the installation verification samples, which are located in prefix.NEW.SDSNSAMP. The installation CLIST tailored and loaded these jobs during your installation or migration to new-function mode.

To edit any of these sample jobs, specify the appropriate fields on the installation panels. Using the installation panels is the only method that you can use to edit these jobs.

Before you run the sample stored procedures, ensure that you have completed one of the following tasks:

- **Configure Db2 for running stored procedures and user-defined functions during installation**
- **Configure Db2 for running stored procedures and user-defined functions during migration**

**Related concepts:**
- Installation verification phases and programs

**Stored procedure sample without result set**

Db2 provides a sample stored procedure that uses the instrumentation facility interface (IFI) to issue Db2 commands. The sample stored procedure also returns the responses to the calling program in a parameter.

This sample application consists of two jobs: DSNTEJ6S and DSNTEJ6P. These jobs are part of the installation verification samples, which are located in prefix.NEW.SDSNSAMP. The installation CLIST tailored and loaded these jobs during your installation or migration.

**Requirements:** You must satisfy the following requirements when running these jobs:

- Run job DSNTEJ6S before job DSNTEJ6P.
- Ensure that the PL/I product is installed on your client and server systems.
- Ensure that Language Environment is installed.

This application prepares and runs the following programs:

- A stored procedure that uses the IFI to issue Db2 commands.
- A client program that receives Db2 command text and calls the stored procedure to issue the commands. When the program receives the responses from the stored procedure in a parameter that is passed back, it prints the results.

For concurrent installations at two Db2 locations, perform the following actions:

- Run the server version of DSNTEJ6S on the server system only; do not run the client version of DSNTEJ6S on the server.
- Run the client version of DSNTEJ6P on the client system only; do not run the server version of DSNTEJ6P on the client.

**Job DSNTEJ6S:**

Job DSNTEJ6S compiles and link-edits the sample stored procedure DSN8EP2. DSNTEJ6S also updates the SYSIBM.SYSCOLUMN catalog table with information about the stored procedure.

The installation CLIST edits this job only if the following conditions are met:
You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.

You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

You specified a name in the REMOTE LOCATION field of panel DSNTIPY. The name can match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR.

If you have SQL statements in your stored procedure, you must remove the comment character in the JCL from the step that binds the stored procedure package.

You must run job DSNTEJ6S on the server Db2 subsystem.

If DSNTEJ6S runs successfully, it produces the return codes that are shown in the table below.

Table 101. DSNTEJ6S return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06SS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06SS02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06SS03</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:

1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

If SQLCODE -592 is received during the execution of DSNTEJ6T, ensure that SPA is active.

Job DSNTEJ6P:

Job DSNTEJ6P compiles, link-edits, binds, and runs a sample program, DSN8EP1, which calls the sample stored procedure that is created by job DSNTIJ6S.

The sample caller first connects to the location that is specified in the REMOTE LOCATION field of panel DSNTIPY. If the remote location name is the same as the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, the local Db2 subsystem is used as both the client and server.

The installation CLIST edits this job only if the following conditions are met:

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.
- You specified a name in the REMOTE LOCATION field of panel DSNTIPY. The name can match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR.

Before you run DSNTEJ6P, ensure that DSNTEJ6S has been run on the server Db2 subsystem to create the sample stored procedure.
If DSNTEJ6P runs successfully, it produces the following return codes.

Table 102. DSNTEJ6P return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06PS01</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PLKED?</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06PS02</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH06PS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

Output from a successful execution of DSNTEJ6P lists each Db2 command that was executed, followed by the messages that were generated by the Db2 command processor.

You can compare the output from this job with the sample output for DSNTEJ6P, which is in member DSN8TJ6P of your prefix.SDSNIVPD data set.

Stored procedure sample with result set

Db2 provides a sample stored procedure that uses the instrumentation facility interface (IFI) to issue Db2 commands. The sample stored procedure also returns the responses to the calling program in a result set.

This sample application consists of two jobs: DSNTEJ6T and DSNTEJ6D. These jobs are part of the installation verification samples, which are located in prefix.NEW.SDSNSAMP. The installation CLIST tailored and loaded these jobs during your installation or migration.

Requirements: You must satisfy the following requirements when running these jobs:
- Run job DSNTEJ6T before job DSNTEJ6D.
- Ensure that C for z/OS is installed.
- Ensure that Language Environment is installed.

This application prepares and runs the following programs:
- A stored procedure that uses the IFI to issue Db2 commands.
- A client program that receives Db2 command text and calls the stored procedure to issue the commands. When the program receives the responses from the stored procedure in a result set, it prints the results.

For concurrent installations at two Db2 locations, perform the following actions:
- Run the server version of job DSNTEJ6T on the server side only; do not run it on the client side.
- Run the client version of job DSNTEJ6D on the client side only; do not run it on the server side.
Job DSNTEJ6T:

Job DSNTEJ6T registers, prepares, and binds the sample stored procedure DSN8ED2 on the server. DSNTEJ6T also defines a created temporary table to receive the IFI output that is returned as a result set.

The installation CLIST edits this job only if the following conditions are met:

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.
- You specified a name in the REMOTE LOCATION field of panel DSNTIPY. The name can match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR.

You must run job DSNTEJ6T on the server Db2 subsystem.

If DSNTEJ6T runs successfully, it produces the return codes that are shown in the table below.

<table>
<thead>
<tr>
<th>Table 103. DSNTEJ6T return codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>PH06TS01</td>
</tr>
<tr>
<td>PH06TS02</td>
</tr>
<tr>
<td>PH06TS03</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>PH06TS04</td>
</tr>
</tbody>
</table>

1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

If SQLCODE -592 is received during the execution of DSNTEJ6T, ensure that SPA is active.

Job DSNTEJ6D:

Job DSNTEJ6D compiles, link-edits, binds, and runs sample program DSN8ED1, which calls the sample stored procedure that is created by job DSNTEJ6T.

The sample caller first connects to the location that is specified in the REMOTE LOCATION field of panel DSNTIPY. If the remote location name is the same as the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, the local Db2 subsystem is used as both the client and server.

The installation CLIST edits this job only if the following conditions are met:

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.
- You specified a name in the REMOTE LOCATION field of panel DSNTIPY. The name can match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR.
Before you run DSNTEJ6D, ensure that DSNTEJ6T has been run on the server Db2 subsystem to create the sample stored procedure.

If DSNTEJ6D runs successfully, it produces the return codes that are shown in the table below.

Table 104. DSNTEJ6D return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06DS01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0000 or 0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06DS02</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>PH06DS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

Output from a successful execution of DSNTEJ6D lists each Db2 command that was executed, followed by the messages that were generated by the Db2 command processor.

You can compare the output from this job with the sample output for DSNTEJ6D, which is in member DSN8TJ6D of your prefix.SDSNIVPD data set.

Sample callers of utilities stored procedures
You can use the Db2-supplied utilities stored procedures to run Db2 utilities from Db2 application programs by using the SQL CALL statement.

The following utilities stored procedures are supplied with Db2:
- DSNUTILS (deprecated)
- DSNUTILU
- DSNUTILV

When called, the utilities stored procedures perform the following actions:
1. Dynamically allocates the specified data sets
2. Creates the utility input stream (SYSIN)
3. Invokes Db2 utilities (DSNUTILB)
4. Deletes all rows currently in the created temporary table (SYSIBM.SYSPRINT)
5. Captures the utility output stream (SYSPRINT)
6. Puts the output into the created temporary table (SYSIBM.SYSPRINT)

The utilities stored procedures must run as a WLM-managed stored procedure.

Related reference:
- DSNUTILS stored procedure (deprecated) (Db2 SQL)
- DSNUTILU stored procedure (Db2 SQL)
- DSNUTILV stored procedure (Db2 SQL)
Job DSNTEJ6X:

Job DSNTEJ6X compiles, link-edits, binds, and runs sample C program DSN8EDA, which invokes the DSNUTILV stored procedure to execute a utility. DSNTEJ6X executes DSN8EDA twice, first passing the utility control statement to DSNUTILV as a string and then passing it as a LOB file reference.

For convenience on TSO, the utility control statement that is input to DSN8EDA is encoded in EBCDIC and converted to Unicode before being passed to DSNUTILV.

Job DSNTEJ6X uses the local Db2® subsystem as both the client and the server subsystem.

The installation CLIST edits this job only if you specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

Before running job DSNTEJ6X, ensure that DSNUTILV has been created as part of the installation or migration process.

Recommendation: Use the DSNWLM_UTILS core WLM environment for DSNUTILV.

If DSNTEJ6X completes successfully, it produces the return codes that are shown in the following.

Table 105. DSNTEJ6X return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06XS00</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS03</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS04</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS05</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS06</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06XS07</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

Output from a successful execution of DSNTEJ6X lists the parameters specified followed by the messages generated by the Db2 DIAGNOSE DISPLAY MEPL utility.

Related tasks:
- Installation step 22: Set up Db2-supplied routines
- Migration step 24: Set up Db2-supplied routines

Related reference:
- DSNUTILV stored procedure (Db2 SQL)
Job DSNTEJ6U:

Job DSNTEJ6U compiles, link-edits, binds, and runs sample PL/I program DSN8EPU, which invokes the DSNUTILS stored procedure to execute a utility.

Job DSNTEJ6U uses the local Db2 subsystem as both the client and the server subsystem.

The installation CLIST always edits this job.

Before running job DSNTEJ6U, ensure that DSNUTILS has been created as part of the installation or migration process.

Recommendation: Use the DSNWLM_UTILS core WLM environment for DSNUTILS.

If DSNTEJ6U completes successfully, it produces the return codes that are shown in the table below.

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06US01</td>
<td>PPLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLI</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06US02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06US03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the PL/I LE PRELINKER PHASE field is set to NO.

Output from a successful execution of DSNTEJ6U lists the parameters specified followed by the messages generated by the Db2 DIAGNOSE DISPLAY MEPL utility.

You can compare the output from this job with the sample output for DSNTEJ6U found in member DSN8TJ6U in your prefix.SDSNIVPD data set.

Related tasks:
- Installation step 22: Set up Db2-supplied routines
- Migration step 24: Set up Db2-supplied routines

Related reference:
- DSNUTILS stored procedure (deprecated) (Db2 SQL)

Job DSNTEJ6R:

Job DSNTEJ6R compiles, link-edits, binds, and runs sample C program DSN8ED8, which invokes the DSNUTILU stored procedure to execute a utility.

For convenience on TSO, the utility control statement that is input to DSN8ED8 is encoded in EBCDIC and converted to Unicode before being passed to DSNUTILU.
DSN8ED8 can accept either a utility statement or a file reference variable that specifies the name of a data set that contains a utility control statement. The specified data set must be in variable record format.

Job DSNTEJ6R uses the local Db2 subsystem as both the client and the server subsystem.

The installation CLIST edits this job only if you specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

Before running job DSNTEJ6R, ensure that DSNUTILU has been created as part of the installation or migration process.

**Recommendation:** Use the DSNWLM_UTILS core WLM environment for DSNUTILU.

If DSNTEJ6R completes successfully, it produces the return codes that are shown in the table below.

**Table 107. DSNTEJ6R return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06RS00</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH06RS01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06RS02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06RS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**

1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

Output from a successful execution of DSNTEJ6R lists the parameters specified followed by the messages generated by the Db2 DIAGNOSE DISPLAY MEPL utility.

You can compare the output from this job with the sample output for DSNTEJ6U found in member DSN8TJ6U in your prefix.SDSNIVPD data set.

**Related concepts:**

- Utility control statements (Db2 Utilities)

**Related tasks:**

- Installation step 22: Set up Db2-supplied routines
- Migration step 24: Set up Db2-supplied routines

**Related reference:**

- DSNUTILU stored procedure (Db2 SQL)

**Job DSNTEJ6V:**

Job DSNTEJ6V compiles, link-edits, binds, and runs sample C++ program DSN8EE1, which invokes the DSNUTILS stored procedure to execute a utility.
Job DSNTEJ6V uses the local Db2 subsystem as both the client and the server subsystem.

The installation CLIST edits this job only if the following conditions are met:
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.
- You indicated on panel DSNTIPU that an IBM C++ compiler is installed.

Before running job DSNTEJ6V, ensure that DSNUTILS has been created as part of the installation or migration process.

**Recommendation:** Use the DSNWLM_UTILS core WLM environment for DSNUTILS.

If DSNTEJ6V completes successfully, it produces the return codes that are shown in the table below.

**Table 108. DSNTEJ6V return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06VS01</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0004</td>
</tr>
<tr>
<td>PH06VS02</td>
<td>PC1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>CP1</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PC2</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP2</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06VS03</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06VS04</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the CPP LE PRELINKER PHASE field is set to NO.

A successful execution of DSNTEJ6V unloads rows and columns from the PROJ sample table.

You can compare the output from this job with the sample output for DSNTEJ6V found in member DSN8TJ6V in your prefix.SDSNIVPD data set.

**Related tasks:**
- Installation step 22: Set up Db2-supplied routines
- Migration step 24: Set up Db2-supplied routines

**Related reference:**
- DSNUTILS stored procedure (deprecated) (Db2 SQL)

**Job DSNTEJ6W**

Job DSNTEJ6W is a JCL job that creates and initializes a sample SAF resource profile. DSNTEJ6W also prepares, binds, and executes DSN8ED6.
DSN8ED6 is a C language caller of the WLM_REFRESH stored procedure. It accepts the WLM environment name and, optionally, the subsystem ID and an SQLID to be in effect when WLM_REFRESH is invoked.

If a name is specified in the WLM ENVIRONMENT field on installation panel DSNTIPX, the installation CLIST customizes DSNTEJ6W to run in and recycle that WLM environment.

DSNTEJ6W also includes commented-out job steps that show:
- How to prepare the DSNTWR external module from the DSNTWRS sample source code.
- How to prepare the DSNTWRE module from the DSNTWRE sample source code. DSNTWRE is used by DSNTWR.

Do not uncomment and run these jobs unless you maintain a customized copy of DSNTWRS or DSNTWRE. If you do not customize the source code, use the Db2-supplied DSNTWR and DSNTWRE modules in the prefix.SDSNLOAD library and the package that is bound by installation job DSNTIJRT.

Job DSNTEJ6W uses the local Db2 subsystem as both the client and the server subsystem.

Before running job DSNTEJ6W, ensure that WLM_REFRESH has been created as part of the installation or migration process. If you are not authorized to create special resource profiles, have your system security administrator perform the first step of this job. The SQLID that is used to run DSNTEJ6W needs READ access to this profile.

If DSNTEJ6W completes successfully, it produces the return codes that are shown in the table below.

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06WS00</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06WS01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06WS02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06WS03</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKEII</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06WS04</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06WS05</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>
Table 109. DSNTEJ6W return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Return codes for PH06WS00, PH06WS01 and PH06WS02 are generated only if the job steps are uncommented.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related tasks:
- Installation step 22: Set up Db2-supplied routines
- Migration step 24: Set up Db2-supplied routines

Related reference:
- WLM_REFRESH stored procedure (Db2 SQL)

Job DSNTEJ6Z

Job DSNTEJ6Z generates a report of current subsystem parameter settings. This report is generated by DSN8ED7, a C-language caller of stored procedure ADMIN_INFO_SYSPARM. You must have MONITOR1 privileges to run ADMIN_INFO_SYSPARM.

Job DSNTEJ6Z uses the local Db2 subsystem as both the client and the server subsystem.

The installation CLIST edits this job only if the following conditions are met:
- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

Before running job DSNTEJ6Z, ensure that ADMIN_INFO_SYSPARM has been created as part of the installation or migration process.

If DSNTEJ6Z completes successfully, it produces the return codes that are shown in the table below.

Table 110. DSNTEJ6Z return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06ZS01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td>C</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>PLKED</td>
<td>0004</td>
<td></td>
</tr>
<tr>
<td>LKED</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>PH06ZS02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06ZS03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

A successful execution of DSNTEJ6Z provides a report as shown in the figure below.
Related tasks:

Installation step 22: Set up Db2-supplied routines
Migration step 24: Set up Db2-supplied routines

Sample ODBA stored procedure

IMS Open Database Access (ODBA) support enables a Db2 stored procedure to directly connect to an IMS DBCTL system and issue DL/I calls to access IMS databases.

This sample ODBA stored procedure application consists of two jobs: DSNTEJ61 and DSNTEJ62. These jobs are part of the installation verification samples, which are located in prefix.NEW.SDSNSAMP. The installation CLIST tailored and loaded these jobs during your installation or migration.

Requirements: You must satisfy the following requirements when running these jobs:

- Run job DSNTEJ61 before DSNTEJ62.
- Ensure that Enterprise COBOL for z/OS is installed.
- Ensure that Language Environment is installed.
- Start a WLM-established stored procedure address space to run DSNTEJ61 and DSNTEJ62. You need to update the startup procedure for the WLM-established stored procedure address space to add the ODBA data set names to the STEPLIB and DFSRESLIB concatenations. An example of a data set name for ODBA is IMSVS.RESLIB.

Related tasks:

- Assigning stored procedures and functions to WLM application environments
- Writing an external procedure to access IMS databases (Db2 Application programming and SQL)

Job DSNTEJ61:

Job DSNTEJ61 prepares sample stored procedure DSN8EC1, which uses ODBA. DSN8EC1 can add, update, delete, and display telephone directory records from the IMS sample database, DFSIVD1. DSN8EC1 shows how the AERTDLI API is used to issue IMS DL/I calls.

The installation CLIST edits this job only if the following conditions are met:

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.
You specified a name in the REMOTE LOCATION field of panel DSNTIPY. The name can match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR.

Before running job DSNTEJ61, read the dependencies information of the job prolog to verify that the server site is correctly configured and has a connection to an IMS subsystem.

If DSNTEJ61 runs successfully, it produces the return codes that are shown in the table below.

Table 111. DSNTEJ61 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH061S01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH061S02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH061S03</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.

Related concepts:
Job DSNTEJ62

Job DSNTEJ62:

Job DSNTEJ62 prepares and invokes the sample client program DSN8EC2, which calls stored procedure DSN8EC1 multiple times to add, delete, and display telephone directory records. DSN8EC1 is created by job DSNTEJ61.

The sample caller first connects to the location that is specified in the REMOTE LOCATION field of panel DSNTIPY. If the remote location name is the same as the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, the local Db2 subsystem is used as both the client and server.

The installation CLIST edits this job only if the following conditions are met:
• You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
• You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.
• You specified a name in the REMOTE LOCATION field of panel DSNTIPY. The name can match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR.

Before running DSNTEJ62, complete the manual editing that is described in the dependencies information in the job prolog.

If DSNTEJ62 runs successfully, it produces the return codes that are shown in the table below.
Table 112. DSNTEJ62 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH062S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH062S02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH062S03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.

Related concepts:
- Job DSNTEJ61

Sample SQL procedures
Db2 for z/OS provides applications for external SQL procedures and native SQL procedures.

The applications for external SQL procedures are:
- DSNTEJ63
- DSNTEJ64
- DSNTEJ65
  - C and Language Environment are required for jobs DSNTEJ63, DSNTEJ64, and DSNTEJ65.

The applications for native SQL procedures are:
- DSNTEJ66

Related tasks:
- Assigning stored procedures and functions to WLM application environments (Db2 Performance)

Job DSNTEJ63:
Job DSNTEJ63 prepares the sample external SQL procedure, DSN8ES1, which accepts a department number and returns salary and bonus data.

The installation CLIST edits this job only if the following conditions are met:
- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

If a remote location was specified in the REMOTE LOCATION field of panel DSNTIPY, job DSNTEJ63 should be run on the remote server subsystem.

Before running DSNTEJ63, you should manually edit the job as described in the dependencies information in the job prolog.
If DSNTEJ63 runs successfully, it produces the return codes that are shown in the following table.

Table 113. DSNTEJ63 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH063S01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH063S02</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>PCC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH063S03</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

Job DSNTEJ64:

Job DSNTEJ64 prepares and executes DSN8ED3, which is a sample routine that calls the sample external SQL procedure, DSN8ES1. DSN8ES1 was created by job DSNTIJ63.

The sample caller first connects to the location that is specified in the REMOTE LOCATION field of panel DSNTIPY. If the remote location name is the same as the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, the local Db2 subsystem is used as both the client and server.

The installation CLIST edits this job only if the following conditions are met:
- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

Before running job DSNTEJ64, ensure that job DSNTEJ6S was run on the server subsystem. Also run job DSNTEJ63 before running job DSNTEJ64. Before running DSNTEJ64, you should manually edit the job as described in the dependencies information in the job prolog.

If DSNTEJ64 runs successfully, it produces the return codes that are shown in the following table.

Table 114. DSNTEJ64 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH064S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 114. DSNTEJ64 return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH064S02</td>
<td>0000 or 0004</td>
<td></td>
</tr>
<tr>
<td>PH064S03</td>
<td>0000</td>
<td></td>
</tr>
</tbody>
</table>

Note:
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

You can compare the output from this job to the sample output for DSNTEJ64, which is found in member DSN8TJ64 in data set named prefix.SDNSIVPD.

Job DSNTEJ65:

Job DSNTEJ65 prepares and runs a sample application that calls the SYSPROC.DSNTPSMP stored procedure processor to create a sample external SQL procedure. The sample external SQL procedure is created at the location that is specified in the REMOTE LOCATION field of panel DSNTIPY.

Job DSNTEJ65 then prepares and runs a sample application that calls the external SQL procedure. The sample caller first connects to the location that is specified in the REMOTE LOCATION field of panel DSNTIPY.

If a remote location is specified in the REMOTE LOCATION field name on installation panel DSNTIPY, ensure that both Db2 REXX language support and the DSNTPSMP stored procedure are available at that location. If the remote location name is the same as the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, the local Db2 subsystem is used as both the client and server.

The major components of job DSNTEJ65 are as follows:
- DSNWLM_REXX is the WLM core environment for DSNTPSMP.
- DSN8ED4 is a sample C program that calls the Db2 external SQL procedures processor.
- DSN8ES2 is a sample external SQL procedure that calculates employee bonuses.
- DSN8ED5 is a sample C program that calls the external SQL procedure DSN8ES2.

The installation CLIST edits this job only if the following conditions are met:
- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

Before running job DSNTEJ65, ensure that DSNTPSMP has been created as part of the installation or migration process.

Recommendation: DSNTPSMP requires that the default EBCDIC CCSID that is used by Db2 also be compatible with the C compiler. Do not run DSNTEJ65 if the default EBCDIC CCSID for Db2 is not compatible with the C compiler. Using an incompatible CCSID results in compile-time errors. Examples of incompatible CCSIDs include 290, 930, 1026, and 1155.
If DSNTEJ65 runs successfully, it produces the return codes that are shown in the following table.

**Table 115. DSNTEJ65 return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH06SS01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06SS02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06SS03</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06SS04</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06SS05</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH06SS06</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH06SS07</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

You can compare the output from this job to the sample output for DSNTEJ65, which is found in member DSN8TJ65 in the data set named prefix.SDSNIVPD.

**Related tasks:**
- Installation step 22: Set up Db2-supplied routines
- Migration step 24: Set up Db2-supplied routines

**Job DSNTEJ66:**

Job DSNTEJ66 creates a sample native SQL procedure. It also prepares and executes a sample caller of that SQL procedure on the local Db2 subsystem. It then modifies the SQL procedure slightly, activates the change on the local subsystem, and executes the sample caller to call the updated SQL procedure.

Specifically, job DSNTEJ66 completes the following actions:

- Creates a sample native SQL procedure called DSN8.DSN8ES3 that generates and returns a CREATE PROCEDURE statement for a given stored procedure.
- Prepares and executes a sample caller of DSN8ES3 called DSN8ED9.
- Illustrates how to use ALTER PROCEDURE... ADD VERSION to create a version V2 of DSN8ES3 that does the same thing as the original version but also adds a terminating semicolon at the end of the generated CREATE PROCEDURE statement.
- Illustrates how to use ALTER ACTIVATE to make version V2 the active version of DSN8ES3.
- If you specify a remote Db2 subsystem in the REMOTE LOCATION field and it does not match the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, additional job steps do the following:
  - Deploys the sample native SQL procedure on the remote Db2 subsystem.
Binds the sample caller on the remote Db2 subsystem and execute it to call the native SQL procedure at that site.

If the REMOTE LOCATION field is blank or matches the name that is specified in the DB2 LOCATION NAME field of panel DSNTIPR, the deploy-related steps are removed from DSNTEJ66 by the installation CLIST.

As part of the setup for deploying DSN8ES3 on a remote server, the DSNTEP2 application needs to be able to connect to the remote site.

The installation CLIST edits this job only if the following conditions are met:

- You specified AUTO or COMMAND in the DDF STARTUP OPTION field of panel DSNTIPR.
- You specified a name in the WLM ENVIRONMENT field of panel DSNTIPX.

Before running DSNTEJ66, perform the manual editing described in the dependencies information in the job prolog. You must also manually tailor DSN8WLMP, the sample WLM startup procedure for DSNTPSMP. If DSNTEJ66 runs successfully, it produces the return codes that are shown in the table below.

**Table 116. DSNTEJ66 return codes**

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH066S01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S03</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S04</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH066S05</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH066S06</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S07</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S08</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S09</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S10</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S11</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S12</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH066S13</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S14</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH066S15</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH066S16</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**

1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

You can compare the output from this job to the sample output for DSNTEJ66, which is found in member DSN8TJ66 in the data set named prefix.SDSNIVPD.
Job DSNTEJ67:

Job DSNTEJ67 shows how to convert sample external SQL procedure DSN8.DSN8ES2 to a sample native SQL procedure called DSN8.DSN8ES2. The conversion is performed by the Db2 precompiler and various Db2 sample REXX services that are provided in member DSN8ERL1.

Before running job DSNTEJ67, you must successfully run job DSNTEJ65 to create DSN8.DSN8ES2 through the Db2 SQL procedure processor, DSNTPSMP. See the job DSNTEJ67 prolog for information about adapting this job to convert other SQL procedures that are created through DSNTPSMP.

Table 117. DSNTEJ67 return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH067S00</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH067S01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH067S02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH067S03</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH067S04</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

Phase 7: Accessing LOB data

Phase 7 of the installation or migration verification process is an optional phase that demonstrates how to set up and use a Db2 LOB application. This phase creates an extension to the Employee sample database to manage employee resumes and photo images.

You run these jobs in this phase:

- **DSNTEJ7**: Creates and loads the employee resume and photo table.
- **DSNTEJ71, DSNTEJ76**: Uses LOB locators to populate the photo images, then validates that the resume and photo data is stored correctly.
- **DSNTEJ73, DSNTEJ77**: Prepares an ISPF application for viewing employee resume data.
- **DSNTEJ75, DSNTEJ78**: Prepares a GDDM application for viewing employee photo images.

Jobs DSNTEJ75 and DSNTEJ78 are not tailored by the installation CLIST unless you specify non-blank values for GDDM MACLIB and GDDM LOAD MODULES on panel DSNTIPW.

After you run these jobs, you can use ISPF and GDDM to view the sample employee resume and photo data.

**Job DSNTEJ7**

Job DSNTEJ7 demonstrates how to create a LOB table with all the accompanying LOB table spaces, auxiliary tables, and indexes. DSNTEJ7 also demonstrates how to use the Db2 LOAD utility to load a LOB table.

Job DSNTEJ7 consists of the following steps:
<table>
<thead>
<tr>
<th>Step</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drops sample LOB objects</td>
</tr>
<tr>
<td>2</td>
<td>Creates sample Employee Resume and Photo LOB table</td>
</tr>
<tr>
<td>3</td>
<td>Creates aliases for the table, then grants access to it</td>
</tr>
<tr>
<td>4</td>
<td>Uses the Db2 LOAD utility to populate the table</td>
</tr>
<tr>
<td>5</td>
<td>Generates run time statistics on the table</td>
</tr>
</tbody>
</table>

If DSNTEJ7 runs successfully, it produces the return codes that are shown in the table below.

**Table 118. DSNTEJ7 return codes**

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH07S01</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH07S02</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH07S03</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH07S04</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH07S05</td>
<td>DSNUPROC</td>
<td>0000</td>
</tr>
<tr>
<td>PH07S06</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH07S07</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Job DSNTEJ71 (optional)**

Job DSNTEJ71 compiles, link-edits, and binds two sample applications that manipulate LOB data.

The DSN8DLPL sample application demonstrates how to use LOB locators to populate a LOB column. The DSN8DLTC sample application validates the contents of the LOB table, verifying that it was populated correctly.

In previous releases, you had to run job DSNTEJ71 to populate LOB columns that were greater than 32 KB. In Db2 12, the LOAD utility populates all columns of the sample LOB table, so job DSNTEJ71 is optional.

If DSNTEJ71 runs successfully, it produces the return codes that are shown in the table below.

**Table 119. DSNTEJ71 return codes**

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH071S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH071S02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH071S03</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 119. DSNTEJ71 return codes (continued)

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH071S04</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH071S05</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**

1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

You can compare the output from this job with the sample output for DSNTEJ71 found in member DSN8TJ71 in your prefix.SDSNVPD data set.

**Job DSNTEJ73**

Job DSNTEJ73 compiles the DSN8DLRV sample application. DSN8DLRV demonstrates how to use built-in functions like POSSTR and SUBSTR to traverse a CLOB column and extract data from it.

DSN8DLRV is written in C language and linked with ISPF and Db2 Call Attach Facility. The package name and plan name are both DSN8LRvr, where vr is the Db2 version and release.

If DSNTEJ73 runs successfully, it produces the return codes that are shown in the table below.

Table 120. DSNTEJ73 return codes

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH073S01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PLKED⁴</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>LKED</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH073S02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PLKED⁴</td>
<td></td>
<td>0004</td>
</tr>
<tr>
<td>LKED</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH073S03</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

**Note:**

1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

**Related tasks:**

*Starting an application in an ISPF/TSO environment in phase 7*

**Job DSNTEJ75**

Job DSNTEJ75 runs sample program DSN8DLPV, which demonstrates how to manipulate BLOB data (employee photo images).

This job is not edited by the CLIST unless the GDDM MACLIB and GDDM LOAD MODULES fields on panel DSNTIPW are non-blank.
DSN8DLPV is a C language program that is linked with ISPF, GDDM and the Db2 Call Attach Facility. The package name and plan name are both DSN8LP<sup>v</sup>r, where <sup>v</sup>r is the Db2 version and release. To run DSN8DLPV you must include the GDDM load module library (SADMMOD) in the logon procedure or in the ISPLLIB concatenation.

If DSNTJEJ75 runs successfully, it produces the return codes that are shown in the table below.

### Table 121. DSNTJEJ75 return codes

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH075S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH075S02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the C LE PRELINKER PHASE field is set to NO.

### Related tasks:

- **Starting an application in an ISPF/TSO environment in phase 7**

#### Job DSNTJEJ76

Job DSNTJEJ76 compiles, link-edits, and binds two sample COBOL applications that manipulate LOB data.

The DSN8CLPL sample application demonstrates how to use LOB locators to populate a LOB column that is larger than 32 KB. The DSN8CLTC sample application validates the contents of the LOB table, verifying that it was populated correctly.

If DSNTJEJ76 runs successfully, it produces the return codes that are shown in the table below.

### Table 122. DSNTJEJ76 return codes

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH076S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH076S02</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH076S03</td>
<td></td>
<td>0000 or 0004</td>
</tr>
<tr>
<td>PH076S04</td>
<td></td>
<td>0000</td>
</tr>
<tr>
<td>PH076S05</td>
<td></td>
<td>0000</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.
After running DSNTEJ76, you can compare the output with the sample output for DSNTEJ76 found in member DSN8TJ76 in your prefix.SDSNIVPD data set.

Job DSNTEJ77

Job DSNTEJ77 compiles the DSN8CLRV sample COBOL application. DSN8CLRV demonstrates how to use built-in functions, such as POSSTR and SUBSTR, to traverse a CLOB column and extract data from it.

DSN8CLRV is written in the COBOL language and linked with ISPF and Db2 Call Attach Facility. The package name and plan name are both DSN8CR\(vr\), where \(vr\) is the Db2 version and release.

To run DSN8CLRV, you must run the first step of job DSNTEJ73 to compile DSN8SDM.

If DSNTEJ77 runs successfully, it produces the return codes that are shown in the table below.

**Table 123. DSNTEJ77 return codes**

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH077S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED()</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH077S02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.

Job DSNTEJ78

Job DSNTEJ78 runs sample COBOL program DSN8CLPV, which demonstrates how to manipulate BLOB data (employee photo images).

This job is not edited by the CLIST unless the GDDM MACLIB and GDDM LOAD MODULES fields on panel DSNTIPW are non-blank.

DSN8CLPV is a COBOL language program linked with ISPF, GDDM and the Db2 Call Attach Facility. The package name and plan name are both DSN8CP\(vr\), where \(vr\) is the Db2 version and release. To run DSN8CLPV you must include the GDDM load module library (SADMMOD) in the logon procedure or in the ISPLLIB concatenation.

If DSNTEJ78 runs successfully, it produces the return codes that are shown in the table below.
### Table 124. DSNTEJ78 return codes

<table>
<thead>
<tr>
<th>JOBSTEP</th>
<th>PROCSTEP</th>
<th>Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH078S01</td>
<td>PC</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>COB</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED[1]</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>PH078S02</td>
<td></td>
<td>0000 or 0004</td>
</tr>
</tbody>
</table>

**Note:**
1. The PLKED step does not exist in the job if the value of the COBOL LE PRELINKER PHASE field is set to NO.

Related tasks:
- Starting an application in an ISPF/TSO environment in phase 7

## Starting an application in an ISPF/TSO environment in phase 7

You must have access to ISPF load module libraries to run the Employee Resume and Photo sample applications.

### Procedure

To start the application:
1. Enter a CALL command from option 6 of the ISPF primary option menu.
2. To start the Employee Resume and Photo sample applications, enter the following:

   ```
   CALL 'prefix.RUNLIB.LOAD(DSN8SDM)'
   ```

   After you enter this command, Db2 displays the sample applications panel.

   Choosing option 4 or option 6 on the sample applications panel during Phase 7 invokes the "Employee Resume" sample application, which processes CLOB data. Choosing option 5 or option 7 on the sample applications panel during Phase 7 invokes the "Employee Photo" sample application, which processes BLOB data.

### Related tasks:
- Working with the employee resume and photo scenarios
- Starting an application in an ISPF/TSO environment
- Make DB2I panels, messages, and load modules available to ISPF and TSO

## The sample applications

After you run the installation verification jobs, you can use the sample applications to test and verify your new installation or migration.

The names of the sample applications have changed for Db2 12. Check to make sure you have the authority to run the Db2 12 sample programs.

Brief scenarios describe how to display, update, add, and delete information using the sample applications. Another scenario describes how to view or change information using a combination of organization and project applications. This scenario contains problem-solving exercises based upon creating and staffing a new department with new projects.
The output from the installation verification steps discussed here appears in your prefix.SDSNIVPD data set.

Related tasks:
- Managing explicit privileges (Managing Security)

Printing options for the sample application listings

The source statements that are contained in prefix.SDSNSAMP can be printed using ISPF facilities, IEBPTPCH, or local facilities.

Most of the Db2 sample applications are contained in prefix.SDSNSAMP. The modules making up the SQLCA formatter routine (DSNTIAR, DSNTIAC, DSNTIA1, and DSNTIAM) are not in the prefix.SDSNSAMP library. They are provided in object form in prefix.SDSNLOAD.

You might not want to print all members of prefix.SDSNSAMP because some of the members are large and contain unprintable data. An alternative is to precompile and compile the wanted program by specifying a cross-reference to the precompiler and compiler. This provides a cross-reference for program variables and is current.

The sample application panels

When you run the interactive sample applications, the sample application panels require you to enter information such as the action you want to complete.

You are prompted for the following information when you run the interactive sample applications.

<table>
<thead>
<tr>
<th>Sample Applications</th>
<th>Distributed Sample Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>• MAJOR SYSTEM</td>
<td>• ACTION</td>
</tr>
<tr>
<td>• ACTION</td>
<td>• OBJECT</td>
</tr>
<tr>
<td>• OBJECT</td>
<td>• SEARCH CRITERIA</td>
</tr>
<tr>
<td>• SEARCH CRITERIA</td>
<td>• LOCATION</td>
</tr>
<tr>
<td>• DATA</td>
<td>• DATA</td>
</tr>
</tbody>
</table>

Regard these categories as a family of values that are used together to specify a task that is to be performed. For MAJOR SYSTEM, ACTION, OBJECT, and SEARCH CRITERIA, a character code of one or two characters is used as a form of shorthand to indicate the criteria that you want. The system provides a list of these codes with their meanings. A valid location name of 1 to 16 characters must be used for LOCATION. The value for DATA must be consistent with the data type and length of SEARCH CRITERIA.

MAJOR SYSTEM

MAJOR SYSTEM specifies the major application area. In the sample application, there are two major systems: organization and project. These major systems are implemented in separate transactions to keep the plan sizes reasonable. If you are running the Db2 distributed sample program, organization is the only system; therefore, this criterion is not used.

ACTION

ACTION specifies what you want to do with the object (specified on another line of the panel). You can display, update, add (insert), or erase (delete) information about the specified object.
OBJECT

OBJECT specifies the object about which you want information. Normally, the action is associated with the object. Examples of objects are information about an employee (EM) or information about the relationship among departments (DS).

Objects can be specified with the following codes for the organization application:

DE Department—general department and manager information for department specified
DS Department structure—hierarchy information for department specified
EM Employee—information concerning employee specified.

Objects can be specified with the following codes for the project application:

PS Project structure—information on projects and subprojects
AL Activity listing—information concerning the different activities that makes up a project
PR Project—general project information
AS Activity staffing—information about the employees staffed for activities of specified projects
AE Activity estimate—information concerning the estimated staffing and time requirements of specified projects.

You are able only to add, update, or erase information about the selected object, although you can search and display based on other criteria. Items that are added or updated can be changed on the screen. Other fields are protected.

SEARCH CRITERIA

SEARCH CRITERIA helps to locate the specific item of information upon which to act.

The following codes can be specified for the search criteria field for the organization application:

DI Department number
DN Department name
EI Employee number
EN Employee name
MI Manager number
MN Manager name.

The following codes can be specified for the search criteria for the project application:

DI Department number
DN Department name
EI Employee number
EN Employee name
PI Project number
PN Project name
RI Responsible person number
RN Responsible person name.

LOCATION
LOCATION is used only for the distributed application. It describes the location where the action is to take place. If this criterion is left blank, then the local location is assumed.

DATA
DATA further identifies the search criteria target. The data value specified must be consistent with the data type and length of the search criteria code. If the search criterion is an employee name (EN), manager name (MN), or responsible person name (RN), the value of data must be a person’s last name.

Data values can be specified by using either primary selection or secondary selection. Primary selection is the data value itself. Only one set of data values fulfills the request. Secondary selection allows multiple sets of data values to fulfill the request. A brief summary of the sets of data values appear on the screen. Each summary has an associated line number. To display additional information about a certain line, enter the line number in the DATA field. Secondary selection allows the application to display a set of values and then provides a prompt to select a specific DATA value. For example, you can display information about a department (DE) (the OBJECT) with a department number (DI) (the SEARCH CRITERIA) with a DATA value of D11.

Related concepts:
- DATA field values on the sample application panels

Related tasks:
- Connecting distributed database systems

Allowable combinations of values on the sample application panels
The codes on the sample application panels cannot be combined indiscriminately.

For instance, manager number (MI) is a valid search criterion for a department (DE), but employee number (EI) and project number (PI) cannot be used to locate a department.

You can retrieve data by having the panels prompt you for the proper values. It is not necessary to enter the values one line at a time. If you already know all the values you want, they can be entered at the same time. If the values are only partially entered, you must start with ACTION and enter each value in sequence, not skipping over any values. For example, if you know all the values except OBJECT, only ACTION can be entered. You are prompted for OBJECT. Then you can enter OBJECT, SEARCH CRITERIA, and DATA.

DATA field values on the sample application panels
When you run the interactive sample applications, you must use the sample application panels to enter information about the task that you want to complete. Information that is entered in the DATA field on the sample application panels is used with the SEARCH CRITERIA field to define a search.

The values available for DATA are not limited to a select few as are the values for ACTION and OBJECT. There is a wider choice of DATA values and a variety of ways to express them.
If you know only part of a DATA value (for example, you know the department number begins with D), you can specify it as a pattern. The pattern can contain any character string with a special meaning, such as:

- The underscore character, _, represents any single character.
- The percent character, %, represents any string of zero or more characters.

These two special characters can be used in conjunction with other characters to specify a DATA value. The following table demonstrates three ways to use these characters to create a DATA value.

<table>
<thead>
<tr>
<th>Data Value</th>
<th>Search Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%SMITH%</td>
<td>EN (Employee name)</td>
<td>Searches for any last name that contains the word SMITH; for example, BLACKSMITH, SMITHSONIAN, or NESMITHA</td>
</tr>
<tr>
<td>E_1</td>
<td>DI (Department number)</td>
<td>Searches for any department number with E in position 1 and 1 in position 3; for example, E71, E21, or EB1</td>
</tr>
<tr>
<td>%</td>
<td>Any</td>
<td>All values qualify</td>
</tr>
</tbody>
</table>

The values entered on the SEARCH CRITERIA and DATA fields can choose only one item to be displayed. However, the more usual case is that several items are displayed as a list. When this is the case, a secondary selection can be made by choosing the line number of the item of interest.

**Function keys on the sample application panels**

The bottom line of each sample application panel displays the function keys that are active for that panel.

**Function key 2—Resend:** If the panel is blanked out (for example, you pressed the CLEAR key) or you want to refresh the panel, press function key 2 to return (resend) the display you were viewing to the terminal.

**Function key 3—End:** To terminate the application, press function key 3 to clear the screen and continue with other transactions.

**Function key 8—Next:** Sometimes a display of information is too large to fit on one panel. Press function key 8 to scroll forward (the lines move upward).

**Function key 10—Left:** Press function key 10 to move the field of vision up one level in the department structure. For instance, in the organization application, Department E01 is shown on the left, and its subdepartments are shown on the right. When you press function key 10, the screen scrolls so that Department E01 is moved from the left side of the panel to the right side and the department to which it reports appears on the left. All other departments that report to the department now on the left also appear along with Department E01 on the right. Function key 10 performs this function only for IMS and CICS samples.

**Related tasks:**

- Working with the organization application scenario

---

**Working with scenarios**

Work with the scenarios for using the sample applications. The method for invoking these applications depends on the environment that you are working in.
About this task

When an application runs, many areas on the display panel might be highlighted. The data that you enter might not be highlighted, depending on the type of panel that is displayed.

Related tasks:
- Starting an application in an ISPF/TSO environment in phase 6
- Starting an application in a CICS environment
- Starting an application in an IMS environment
- Starting an application in an ISPF/TSO environment

Working with the project application scenario

The project application scenario demonstrates how to view details of a project and update activities that are associated with a project.

About this task

For example, you can find the person responsible for a project and list the activities that are assigned to one of its subprojects. Phase 4 (IMS) and Phase 5 (CICS) prepare the programs that you execute.

Procedure

To view details of a project with the sample application:

Enter the following values on the first panel of the project application:

- On the MAJOR SYSTEM line, enter P for project.
- On the ACTION line, enter D for display.
- On the OBJECT line, enter PS for project structure.
- On the SEARCH CRITERIA line, enter PI for project ID.
- On the DATA line, enter MA2100 as the project ID.

The following example panel shows the selected project and its corresponding subprojects.

![Project Application Panel]

Figure 96. Project application—viewing a project structure
**Updating an activity**
You can use the project application to update an activity information for a project, such as the estimated start date.

**About this task**
Suppose that you want to update activity information for a project with ID IF1000.

**Procedure**
To update the activity:

1. Enter the following values:
   - On the MAJOR SYSTEM, enter `P` for project.
   - On the ACTION line, enter `U` for update.
   - On the OBJECT line, enter `AE` for activity estimate.
   - On the SEARCH CRITERIA line, enter `PI` for project ID.
   - On the DATA line, enter `IF1000` as the project ID.

2. Press the ENTER key, and a list of project IF1000 activities appears on the panel.

3. Choose the activity to be updated. For instance, if you want to update the first activity listed, enter 1 as the DATA value and press the ENTER key. The next panel shows information about the estimated mean staffing requirements of this activity as well as the start and completion dates.

4. To change information about the estimated end date, enter data over the existing information displayed on that input line.

5. After you have verified the change, press ENTER. The next panel displays the updated information, as shown in the following figure:

![Project application—changes accepted](image)

**What to do next**
If you are using CICS, clear the screen and enter a new transaction code. If you are using IMS, clear the screen and enter a new transaction code or a `/FORMAT` command.
Working with the organization application scenario

The organization application scenario shows how to display a list of departments within an organization and the structure of one of these departments.

About this task

This application is executed in Phase 4 for IMS and Phase 5 for CICS.

Procedure

To display a list of departments and the structure of one of the departments:

1. After you enter the appropriate transaction code, you see the first panel of the project application. Enter the following values:
   - On the MAJOR SYSTEM, enter O for organization.
   - On the ACTION line, enter D for display.
   - On the OBJECT line, enter DS for department structure.
   - On the SEARCH CRITERIA line, enter DI for department number.
   - On the DATA line, enter %, which enables you to display a list of all the departments.

   Each department entry is numbered on the far left side of the panel as shown in the figure below.

   ![Figure 98. Organization application—viewing a list of departments](image)

2. To retrieve further information, specify a line number as a data value. This method is called secondary selection. Secondary selection provides prompts to aid in finding the information to be displayed, added, erased, or updated. If only one entry possibility exists, secondary selection is not offered. For example, to view the structure of Department E01, specify a data value of 7 on the DATA entry line (7 is the line number of the entry for Department E01). The result of entering the data value of 7 is a display of Department E01 and its departments as shown in the figure below. The department manager for E01 is listed on the left, and the departments of E01 are listed on the right. Employees of E01 are listed below the subdepartments of E01.
Starting a new operation

You can use the organization application to start a new operation such as displaying or updating a department structure.

Procedure

To start a new operation:

Choose one of the following actions:

- Move the cursor to the D on the ACTION line and retain the D or change it to a different action (add, erase, or update). Follow the displayed options to perform your selected action.
- Alternatively, you can leave the organization application by pressing the PF3 key. If you are using CICS, enter the transaction code. If you are using IMS, clear the screen and enter the /FORMAT command to select the project application. In either case, to proceed with a different operation, select a different ACTION, OBJECT, and so forth.

Adding a new department

You can use the organization application to add a new department and specify information about the department.

Procedure

To add a department to the organization:

1. Start the organization application and enter the following values:
   - On the MAJOR SYSTEM line, enter O for organization
   - On the ACTION line, enter A for add (insert)
   - On the OBJECT line, enter DE for the department that is to be added
   - On the SEARCH CRITERIA line, enter DI for department ID
   - On the DATA line, enter C1, the specific department number

2. Next, you can enter the details of the new department. The four department fields are department number, department name, manager number, and administration department number. Enter the following values:
   - INFORMATION SERVICES for department name
   - 000130 for manager number

Figure 99. Organization application—viewing a department structure
• C01 for the administration department number

Press ENTER to display the panel shown in the figure below. The panel shows the successful addition of the new department.

![Figure 100. Organization application—adding a department](image)

### Related tasks:
- Starting a new operation

### Deleting an entry

You can use the organization application to delete an entry from the department table.

### About this task

Deleting an entry from the department table is a function of the organization major system.

### Procedure

To delete an entry from the department table:

1. Replace the following values on the panel that is on your screen:
   - On the MAJOR SYSTEM line, enter O for organization.
   - On the ACTION line, enter E for erase.
   - On the OBJECT line, enter DE for department.
   - On the SEARCH CRITERIA line, enter DI for department ID.
   - On the DATA line, enter C11 for department name.

2. Press ENTER to display the panel shown in the figure below.
3. Press ENTER again to verify the erase action. The following message appears on the panel:

DSN8013I csect DEPARTMENT SUCCESSFULLY ERASED

Related tasks:
Starting a new operation

Transferring an employee
The procedure for transferring one employee to another department and replacing that employee involves several steps.

About this task
In this scenario, John B. Geyer (manager of the department for Support Services) is transferred to the staff of Spiffy Computer Service Division. Bruce Adamson is assigned as manager of Support Services.

To move Adamson into his new position as manager of Support Services, you must determine his employee number. Transferring an employee is a function of the organization major system.

Procedure
To transfer an employee to another department:
1. Start the organization application and enter the following values:
   • On the MAJOR SYSTEM, enter O for organization.
   • On the ACTION line, enter D for display.
   • On the OBJECT line, enter EM for employee.
   • On the SEARCH CRITERIA line, enter EN for employee name.
   • On the DATA line, enter ADAMSON as the specific employee name.
2. Press ENTER to display the panel showing that Adamson’s employee number is 000150.
3. The next step is for you to change the manager number for the Support Services department to Adamson’s number, 000150. But first you must find the Support Services department. To do this, change ACTION to U (update), OBJECT to DE (department), and SEARCH CRITERIA to DN (department name). Change DATA to %SUPPORT% to specify any department with the word SUPPORT in it.
4. Press ENTER, and a list of departments with support in their name is displayed. Support Services has line number 01. Enter this number at DATA. (The leading zero is not needed.)

5. Press ENTER to display the next panel. The only values that can be changed are department name, manager ID, and administration department ID. Enter Adamson’s employee number in the Support Services department after MANAGER ID. At this point, the data on the manager still pertains to Geyer.

6. Press ENTER to display the panel that shows Adamson as manager of Support Services. The work department ID shown (D11) is still Adamson’s old number. To change Adamson’s work department ID, enter EM for OBJECT, enter EL for SEARCH CRITERIA, and change the employee number to 000150 for DATA.

7. Press ENTER to display the employee information on Adamson. Now that information on Adamson can be updated. The fields that can be changed are employee first name, middle initial, last name, and work department ID. Enter the middle initial for Adamson, which was not in the database, and the department number E01. Press ENTER, and the information on Adamson is updated, including his new department number.

8. The final step is to move Geyer to the correct department. Change the SEARCH CRITERIA and DATA to EN and GEYER, respectively. Press ENTER to obtain the next panel. The employee ID, name, and work department ID can be changed on this panel. However, the only change necessary in this case is to change Geyer’s work department ID to his new one, A00. The panel in the figure below shows the completed entry.

9. To terminate the application and return to the beginning of the operation, press the PF3 key.

**Related tasks:**
- Starting a new operation

**Working with the phone application scenario**

The phone application retrieves information from a phone directory and updates employee phone numbers.

**About this task**

The phone application is used in phase 2 (batch mode), phase 3 (CAF), and interactively in phase 4 (IMS) and phase 5 (CICS).
The phone directory consists of data from a combination (join) of the employee table (DSN8C10.EMP) and the department table (DSN8C10.DEPT). This joined view is called VPHONE. The program also uses a second view called VEMPLP to update the employee table, which does not affect a view that joins tables.

The phone application is designed to operate in batch and interactively in ISPF/TSO, IMS, and CICS. The following table describes the environments in which each phone application operates and the language in which each is written.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Language</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPF/TSO</td>
<td>COBOL</td>
<td>DSN8SC3</td>
</tr>
<tr>
<td>ISPF/TSO</td>
<td>PL/I</td>
<td>DSN8SP3</td>
</tr>
<tr>
<td>IMS</td>
<td>PL/I</td>
<td>DSN8IP3</td>
</tr>
<tr>
<td>CICS</td>
<td>PL/I</td>
<td>DSN8CP3</td>
</tr>
<tr>
<td>batch</td>
<td>COBOL</td>
<td>DSN8BC3</td>
</tr>
<tr>
<td>batch</td>
<td>Fortran</td>
<td>DSN8BF3</td>
</tr>
<tr>
<td>batch</td>
<td>PL/I</td>
<td>DSN8BP3</td>
</tr>
<tr>
<td>batch</td>
<td>C</td>
<td>DSN8BD3</td>
</tr>
</tbody>
</table>

Related tasks:
Starting an application in an ISPF/TSO environment

Viewing or changing an employee's telephone number
A couple of panels are used to view or change an employee's telephone number in the sample telephone application. These panels are the same, regardless of whether IMS or CICS is used.

Procedure

To view or change an employee's telephone number:

1. On the first panel of the telephone application, enter the first and last name of the employee whose telephone number you want to view or change. To see an entire listing of employee numbers, put an * next to the LAST NAME input line. If only part of a first or last name is known, use the percent character (%) to qualify the list of names to appear in the directory. For example, entering K% on the LAST NAME input line calls a list of the telephone numbers of all employees whose last name begins with a K. Similarly, the first name can be qualified.

To keep this sample program as simple as possible and to allow updating, scrolling is not used with the IMS and CICS versions. Scrolling is used with the ISPF/CAF version. Only the first panel of selected names and phone numbers can be displayed. The second panel is the Telephone Directory itself. The employee telephone number is highlighted. To update an employee telephone number, type over the highlighted number and press ENTER. To update a phone number listed under the name Heather A Nicholls, specify NICHOL% when you are not sure if there are one or two Is in Nicholls.
2. Press ENTER to display the panel on which the phone number is highlighted.

3. If you want to change a phone number, type over the number to be changed. You can type over as many numbers as are listed in the display. For example, suppose that you want to change the phone number from 1793 to 1795. After you press ENTER, you get a message confirming the updated phone number. The panel in the following figure shows the updated panel.

--- TELEPHONE DIRECTORY ---

LAST NAME ==> __
FIRST NAME ==> __

LAST NAME  * FOR LIST OF ENTIRE DIRECTORY
FIRST NAME(OPTIONAL)  % FOR GENERIC LIST

Figure 103. Telephone application—first display

Using the phone application under batch
The sample batch phone applications are provided in Fortran (DSN8BF3), COBOL (DSN8BC3), C (DSN8BD3), and PL/I (DSN8BP3).

Procedure

To update or list an employee phone number:

Create a data set that contains information about the phone number that is to be updated. This data set works in combination with another data set that contains JCL for processing information. The first data set consists of card images in the format that is shown in the following table.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACTION—U for update, L for list</td>
</tr>
<tr>
<td>2</td>
<td>Employee last name</td>
</tr>
<tr>
<td>17</td>
<td>Employee first name</td>
</tr>
<tr>
<td>29</td>
<td>Employee number</td>
</tr>
<tr>
<td>35</td>
<td>New phone number</td>
</tr>
</tbody>
</table>

The ACTION code in this card image indicates whether an employee number is to
be updated (U) or listed (L). When updating an employee phone number, only the employee number and the new phone number are specified in the data set. When listing phone numbers, the last name must be specified. Specifying the first name is optional. The * and % can be used with the ACTION code just as they are used with the panels.

Example

The following figure shows an example of an update data set and a list data set. Each time a number is listed or updated, a new data set is created containing a card image like the one in the following figure. The first card in the data set shows the phone number of employee number 000140 being updated (U) to 6767. The second card shows a list (L) for Heather Nicholls. The last card shows a list (L) of all employees whose first names begin with the letters MAR. The example shows the letters MAR followed by a % in the first name column to indicate that only those employees whose first names begin with MAR are to be listed.

The data set that contains the JCL for processing information is supplied with Db2 and is contained in DSNTEJ2P, which is part of prefix.SD5NSAMP. The figure below shows the data set that contains the JCL with the card image data sets embedded.

```
//PH02PS05 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//REPORT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//CARDIN DD * 
U 0001406767
LNICHOLLS HEATHER
L MAR%
//SYSTSIN DD *
DSN SYSTEM(DSN) 
RUN PROGRAM(DSNBBP3) PLAN(DSN8BP12) LIB('prefix.RUNLIB.LOAD') 
END
```

Figure 106. The job control language data set

The complete data set can be submitted to the system either through a card reader or from a terminal through TSO. The figure below contains an example of the batch output.
Working with the distributed organization application scenario

The distributed organization application scenario shows how to work with department and employee data in a distributed environment.

About this task

The scenario shows how to display a department structure, display department information, and update a department at a local location. It also shows how to erase and add an employee at a remote location. This application is executed in Phase 6. The application accesses distributed data with DRDA access.

The department information (DEPT table) is shared by all locations. If you make changes to DEPT table at one location, the DEPT tables at the other locations are updated at the same time. The employee information (EMP table) is unique to each location, containing only the employees that work at that particular location.

After you enter the appropriate transaction code, you see the first panel of the organization application.

Displaying department structure at the local location

You can use the distributed organization sample application to display department structure at the local location.

Procedure

To display a department structure:

1. Enter the following values in the distributed organization sample application:
   • On the ACTION line, enter D for display.
   • On the OBJECT line, enter DS for department structure.
   • On the SEARCH CRITERIA line, enter DI for department number.
   • On the LOCATION line, leave blank, indicating local location.
   • On the DATA line, enter A00 for department number.

---

**Figure 107. Example of the phone application batch output**

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>INITIAL</th>
<th>PHONE</th>
<th>EMPLOYEE WORK</th>
<th>WORK NUMBER</th>
<th>DEPT DEPT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUINTANA</td>
<td>DOLORES</td>
<td>M</td>
<td>6767</td>
<td>000130</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
</tr>
<tr>
<td>NICHOLLS</td>
<td>HEATHER</td>
<td>A</td>
<td>1793</td>
<td>000140</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
</tr>
<tr>
<td>SCOUTTN</td>
<td>MARILYN</td>
<td>S</td>
<td>1682</td>
<td>000180</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>PEREZ</td>
<td>MARIA</td>
<td>L</td>
<td>9001</td>
<td>000270</td>
<td>D21</td>
<td>ADMINISTRATIVE SYSTEMS</td>
</tr>
</tbody>
</table>

---

Installation and Migration Guide
2. Press the ENTER key. The panel below shows the structure of the department requested.

![DB2 ORGANIZATION APPLICATION]

**Figure 108. Starting the distributed organization application**

3. Press ENTER or END to exit.

**Displaying department information at the local location**

You can use the distributed organization sample application to display department structure at the local location.

**Procedure**

To display department information:

1. Enter the following values in the distributed organization sample application:
   - On the ACTION line, enter D for display.
   - On the OBJECT line, enter DE for department.
   - On the SEARCH CRITERIA line, enter DI for department number.
   - On the LOCATION line, leave blank, indicating local location.
   - On the DATA line, enter A00 for department number.
2. Press the ENTER key. The panel below shows the department information that was requested.

```
<table>
<thead>
<tr>
<th>ACTION</th>
<th>OBJECT</th>
<th>SEARCH CRITERIA</th>
<th>LOCATION</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>de</td>
<td>di</td>
<td>(Blank)</td>
<td>a00</td>
</tr>
</tbody>
</table>
```

Figure 110. Starting the distributed organization application

3. Press ENTER or END to exit.

**Updating a department at the local location**

You can use the distributed organization sample application to update department information at the local location.

**Procedure**

To update department information:

1. Enter the following values in the distributed organization sample application:
   - On the ACTION line, enter u for update.
   - On the OBJECT line, enter de for department.
   - On the SEARCH CRITERIA line, enter di for department number.
   - Leave the LOCATION line blank, indicating local location.
   - On the DATA line, enter %, which displays a list of all the departments.
2. Press the ENTER key. The panel below lists the departments that can be updated. Select the department to be updated by inserting an S in the left margin by the department number.

3. Press the ENTER key. The panel below displays the information relevant to the selected department. Enter the information you want to update on this panel; in this case, enter the name of the department.
4. Press the ENTER key to process the updated information. A message appears on this panel that states the update was successful.

5. Press ENTER to return to the previous panel or END to exit. If you return to the previous panel, you can now select another department to update, or press ENTER or END to exit. The same message appears on this panel, indicating the update was successful.
Adding an employee at a remote location
You can use the distributed organization sample application to add information for employees at remote locations.

Procedure

To add an employee at a remote location:
1. Enter the following values:
   - On the ACTION line, enter a for add.
   - On the OBJECT line, enter em for employee.
   - On the SEARCH CRITERIA line, enter ei for employee number.
   - On the LOCATION line, enter your server location for the remote location.
   - On the DATA line, enter the employee ID of the employee to be added (sj0100 in the example below).
2. Press the ENTER key. The panel below allows you to enter information only about the employee. The department information on the panel is protected. Enter the necessary information about the employee.

```
DB2 ORGANIZATION APPLICATION

ACTION ........: A (ADD) E (ERASE)
               D (DISPLAY) U (UPDATE)

OBJECT ........: D (DEPARTMENT) E (EMPLOYEE)
                D (DEPT STRUCTURE)

SEARCH CRITERIA : D (DEPARTMENT ID) M (MANAGER NAME)
                  D (DEPARTMENT NAME) E (EMPLOYEE ID)
                  M (MANAGER ID)   E (EMPLOYEE NAME)

LOCATION .......: your server location (Blank implies local location)

DATA ............: sj0100

PRESS: ENTER to process   END to exit
```

*Figure 117. Starting the distributed organization application*

3. Press the ENTER key to process or END to exit. If you press the ENTER key, a message appears on the panel indicating that the employee has been added.

```
DB2 ORGANIZATION APPLICATION

ADD AN EMPLOYEE

DEPARTMENT ID : 
NAME : 
MANAGER ID : 
ADMIN DEP ID :
LOCATION : 

EMPLOYEE ID : SJ0100
FIRST NAME : W
MIDDLE INITIAL : 
LAST NAME : WALTERS
WORK DEPT ID : F22

PRESS: ENTER TO PROCESS   END TO EXIT
```

*Figure 118. Employee to be added*
4. Press the ENTER key to return to the selection panel or END to exit.

**Erasing an employee at a remote location**

You can use the distributed organization sample application to erase information for employees at remote locations.

**Procedure**

To erase an employee at a remote location:

1. Enter the following values in the distributed organization sample application:
   - On the ACTION line, enter e for erase.
   - On the OBJECT line, enter em for employee.
   - On the SEARCH CRITERIA line, enter ei for employee number.
   - On the LOCATION line, enter your server location for the remote location.
   - On the DATA line, enter %, which displays a list of all the employees.

---

**Figure 119. Employee successfully added**

**Figure 120. Starting the distributed organization application**

2. Press the ENTER key. The panel below lists the employees that can be erased. Select the employee to be erased by inserting an S in the left margin by the employee ID.
3. Press the ENTER key. The panel below displays the information relevant to the selected employee that is to be erased.

4. Press the ENTER key to erase the employee information. A message appears on the panel stating the employee has been successfully erased. You can now erase another employee or press ENTER or END to exit.
Working with the employee resume and photo scenarios

The LOB sample application extends the existing Db2 sample employee database by adding a new table for storing employee resumes and photographs as CLOB and BLOB entries.

About this task

The supporting JCL, application objects, and input data for this table are provided.

The purpose of the LOB sample application is to:

- Provide an IVP for LOB functions.
- Demonstrate how to:
  - Use DDL to create LOB objects.
  - Use the Db2 LOAD utility to populate LOB columns of 32K bytes or less.
  - Create an application program to populate LOB columns of greater than 32K bytes.
  - Use LOB locators and related functions to manipulate LOBs without materializing the data.

The LOB sample application consists of a batch portion and an optional online portion. The batch portion verifies that LOB objects can be created, populated, and successfully read by using locators and supporting functions. The online portion demonstrates further techniques for using and manipulating LOB data.

The batch portion of the LOB sample application consists of the following jobs: DSNTEJ7, DSNTEJ71, DSNTEJ73, DSNTEJ75, DSNTEJ76, DSNTEJ77, and DSNTEJ78.

The optional online portion of the LOB sample application adds two additional scenarios to the existing three scenarios that are provided in IVP phase 3. These scenarios include the following user activities:

1. Invoking the Db2 sample connection manager to display the Db2 ISPF sample application menu, DSN8SSM.
2. Selecting and viewing sample employee resumes by using option 4 of DSN8SSM.
3. Selecting and viewing sample employee photo images by using option 5 of DSN8SSM.

Application programs for the LOB sample are written in the C and COBOL languages. The resume viewer requires ISPF. The photo viewer requires ISPF and GDDM.

**Related concepts:**
- **Phase 7: Accessing LOB data**

**Using the resume application in the LOB application scenario**
You can view employee resumes with the resume application in the LOB application scenario.

**About this task**

The LOB application begins with the initial panel of the ISPF/CAF application.

**Procedure**

To view employee resumes:

1. From the Db2 Sample Application Menu, DSN8SSM, select one of the following options:
   - If you ran job DSNTEJ73, select option 4.
   - If you ran job DSNTEJ77, select option 6.

   DSN8DLRV and DSN8CLRV show the following panels for selecting and viewing sample employee resumes until you press the END key to exit ISPF.

2. Select from the list of available resumes that are displayed on ISPF panel DSN8SSE.

   The employee serial numbers and names are hard-coded into the panel because the data for this sample is predetermined. The formatted resume is displayed on ISPF panel DSN8SSR:
This panel is designed around the sample data. It is assumed that all resume information for an employee will fit predictably and no handling for special cases is provided. The “Interests” section of the resume is not presented because of space constraints on the panel.

Related tasks:
Starting an application in an ISPF/TSO environment

Using the photo application in the LOB application scenario
You can view employee photos with the photo application in the LOB application scenario.

About this task
The LOB application begins with the initial panel of the ISPF/CAF application. This LOB application requires that you include the GDDM load module library (SADMMOD) in your logon procedure or in the ISPLLIB concatenation.

Procedure
1. From the Db2 Sample Application Menu, DSN8SSM, select one of the following options to look at employee photos through GDDM:
   • If you ran job DSNTEJ75, select option 5.
   • If you ran job DSNTEJ78, select option 8.
DSN8DLPV and DSN8CLPV show the following panel for selecting and viewing sample employee photos until you press the END key to exit ISPF.
2. Select from the list of available photos by that are displayed on ISPF panel DSN8SSE.

Related tasks:
- [Starting an application in an ISPF/TSO environment](#)

Sample LOB table: EMP_PHOTO_RESUME
The LOB sample application uses the EMP_PHOTO_RESUME table. The sample jobs create, load, and manipulate the table.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>EMPNO</th>
<th>EMP_ROWID</th>
<th>PSEG_PHOTO</th>
<th>BMP_PHOTO</th>
<th>RESUME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>CHAR(6)</td>
<td>ROWID</td>
<td>BLOB(500K)</td>
<td>BLOB(100K)</td>
<td>CLOB(5K)</td>
</tr>
<tr>
<td>Description:</td>
<td>Employee number</td>
<td>Row identifier</td>
<td>Employee photo (PSEG)</td>
<td>Employee photo (BMP)</td>
<td>Employee resume</td>
</tr>
<tr>
<td>Values:</td>
<td>000130</td>
<td>######</td>
<td>Delores M. Quintana</td>
<td>Delores M. Quintana</td>
<td>Delores M. Quintana</td>
</tr>
<tr>
<td>Values:</td>
<td>000140</td>
<td>######</td>
<td>Heather A Nicholls</td>
<td>Heather A Nicholls</td>
<td>Heather A Nicholls</td>
</tr>
<tr>
<td>Values:</td>
<td>000150</td>
<td>######</td>
<td>Bruce Adamson</td>
<td>Bruce Adamson</td>
<td>Bruce Adamson</td>
</tr>
<tr>
<td>Values:</td>
<td>000190</td>
<td>######</td>
<td>James H. Walker</td>
<td>James H. Walker</td>
<td>James H. Walker</td>
</tr>
</tbody>
</table>

Edit exit routine

The edit exit routine encodes the salary column to protect the confidentiality of the employee's salary.

The edit exit routine is prepared in Phase 1. It works with the employee table (DSN8C10.EMP) and is written in assembler language.

The name of the edit exit routine is DSN8EAE1. When the employee table (DSN8C10.EMP) is changed by either an update or an add, the edit exit routine encodes the salary amount that goes into the SALARY column. When the SALARY column is read from the employee table, the amount is decoded. The encoding and decoding of the salary column protects the confidentiality of the employee's salary.
Huffman compression exit routine

IBM supplies a sample edit routine that compresses data using the Huffman algorithm.

Before using any data compression routine, understand its limitations and consider tailoring it to your particular table. For the restrictions and concerns that apply to the IBM sample, see the comments provided with the code. The routine is called DSN8HUFF and resides in library prefix.SDSNSAMP. IBM supplies a sample edit routine that compresses data using the Huffman algorithm.

Sample field procedure

A sample field procedure is prepared in Phase 1 of the installation or migration verification process. This procedure causes values in a CHAR(6) column to be ordered in the ASCII sorting sequence.

Dynamic SQL statements: DSNTESA, DSNTESQ

prefix.SDSNSAMP library members DSNTESA and DSNTESQ contain dynamic SQL statements to help verify the success of an installation or migration.

DSNTESA

The SQL statements in DSNTESA are run dynamically by SPUFI. DSNTESA is used in Phase 3 of the verification process.

The first group of statements in DSNTESA create a temporary work file table space and defines a created temporary table. The INSERT statements fill the table with names, midterm scores, and final examination results, and the SELECT statement then does a check of the averages. The UPDATE statements assign a grade according to the formula in the first UPDATE statement: 60% for the final and 40% for the midterm. The next SELECT statement produces the entire table. The ROLLBACK statement removes the table space and the table within it.

The following statements make some administrative queries on the system tables:

- The following SELECT statements find all the plans and packages that are owned by the current user, and the date they were bound.

  SELECT NAME, BINDDATE
  FROM SYSIBM.SYSPLAN
  WHERE CREATOR = USER;

  SELECT COLLID, NAME, VERSION, BINDTIME
  FROM SYSIBM.SYSPACKAGE
  WHERE OWNER = USER;

- The following SELECT statements find the plans and packages that require a bind or rebind before they can be run, and the plans and packages that are automatically rebound the next time they are run.

  SELECT NAME, CREATOR, BINDDATE, VALID, OPERATIVE
  FROM SYSIBM.SYSPLAN
  WHERE OPERATIVE = 'N' OR VALID = 'N';

  SELECT COLLID, NAME, VERSION, BINDTIME, VALID
  FROM SYSIBM.SYSPACKAGE
  WHERE OPERATIVE = 'N' OR VALID = 'N';
• The following SELECT statements find all objects required for the current user’s programs.

```sql
SELECT DNAME, BTYPE, BCREATOR, BNAME
FROM SYSIBM.SYSPLANDEP
WHERE BCREATOR = USER
ORDER BY DNAME, BTYPE, BCREATOR, BNAME;
SELECT DCOLLID, DNAME, BTYPE, BQUALIFIER, BNAME
FROM SYSIBM.SYSPACKDEP
WHERE BQUALIFIER = USER
ORDER BY DCOLLID, DNAME, BTYPE, BQUALIFIER, BNAME;
```

• The second SELECT from SYSTABLES provides information about all the DEPT tables regardless of the owner.

```sql
SELECT *
FROM SYSIBM.SYSTABLES
WHERE NAME = 'DEPT';
```

• The SELECT from SYSCOLUMNS supplies a description of the fields of the DSN8C10.DEPT table. This information can also be provided by DCLGEN, and, within a program, the DESCRIBE statement gives this same information.

```sql
SELECT NAME, COLTYPE, LENGTH, SCALE, NULLS, REMARKS, COLNO
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME = 'DEPT' AND TBCREATOR = 'DSN8610'
ORDER BY COLNO;
```

• The following SELECT statements find the kinds of authority a user can have. Determining which tables a specific user can access is relatively complicated because of the various authorities. If the user has SYSADM authority, any table can be accessed.

```sql
SELECT * FROM SYSIBM.SYSPLANAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSPACKAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSUSERAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSDBAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSCOLAUTH WHERE GRANTEE = USER;
SELECT * FROM SYSIBM.SYSRESAUTH WHERE GRANTEE = USER;
```

• The final four SELECT statements show the tables and views that can be accessed directly by the current user, those that can be accessed using a plan, and those that are accessed using the database authority.

```sql
SELECT TCREATOR, TTNAME, STNAME, GRANTOR
FROM SYSIBM.SYSTABAUTH
WHERE GRANTEE = USER;
SELECT BNAME, BTYPE, GRANTOR, NAME
FROM SYSIBM.SYSPLANAUTH, SYSIBM.SYSPLANDEP
WHERE GRANTEE = USER
AND NAME = DNAME
AND EXECUTEAUTH = 'T' OR BTYPE = 'V';
SELECT DCOLLID, BNAME, BTYPE, BQUALIFIER, BNAME
FROM SYSIBM.SYSPACKAUTH, SYSIBM.SYSPACKDEP
WHERE GRANTEE = USER
AND DCOLLID = DCOLLID
AND NAME = DNAME
AND EXECUTEAUTH = 'T' OR BTYPE = 'V';
SELECT NAME, CREATOR, TYPE, DBNAME, TSNAME
FROM SYSIBM.SYSTABLES
WHERE DBNAME IN
(SELECT NAME FROM SYSIBM.SYSDBAUTH
WHERE GRANTEE = USER
AND DBADMAUTH = 'T');
DSNTESQ contains a set of queries to check consistency between catalog tables.

The SQL statements are in a format available for input to SPUFI and DSNTEP2. If SPUFI is not bound when you want to execute these queries, you can use DSNTEP2 from your previous Db2 version.

Before running these queries, you should run the CHECK DATA and CHECK INDEX utilities.

DSNTESQ contains SQL that creates copies of the catalog using segmented table spaces. In some cases, the queries in DSNTESQ run faster when run on copies of the catalog instead of the actual catalog because the copies have additional indexes. If you plan to use the copies of the catalog, use the comment lines in DSNTESQ for guidance.

**Dynamic SQL programs: DSNTIAD, DSNTEP2, DSNTIAUL**

DSNTIAD, DSNTEP2, and DSNTIAUL are sample dynamic SQL programs and must be compiled, link-edited, and bound as usual.

SPUFI is a part of the distributed product. An installation job is used to bind it. It can be used only with ISPF.

Related concepts:

- Sample applications supplied with (Db2 Application programming and SQL)
Chapter 11. Working with additional capabilities for Db2

After you complete the installation or migration process, you can work with additional capabilities of Db2 such as stored procedures and remote communication.

Installation information for procedures and functions that are supplied with Db2

Db2 supplies a number of stored procedures and user defined functions that job DSNTIJRT defines and binds during installation or migration.

For supplied routines, job DSNTIJRW creates the core WLM environments that are listed in the following table. If you decide to set up your own WLM application environments, use the following table to help you group the supplied routines into the appropriate environments. If your system resources are constrained, you might need to choose a lower NUMTCB value than the recommended one.

For more information about each of the stored procedures that are provided with Db2, see [Procedures that are supplied with Db2 (Db2 SQL)](https://www.ibm.com/docs/en/db2/11.5?topic=procedures-supplied-with-db2-db2-sql).

<table>
<thead>
<tr>
<th>Supplied routine</th>
<th>Schema name</th>
<th>Recommended value of NUMTCB</th>
<th>Core WLM environment</th>
<th>Does the routine have to run in an APF-authorized environment?</th>
<th>Does the routine have to be program-controlled?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIN_COMMAND_DB2</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_COMMAND_DSN</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_REXX</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_COMMAND_MVS</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_COMMAND_UNIX</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_PGM_CONTROL</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_DS_BROWSE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_DS_DELETE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_DS_LIST</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_DS_RENAME</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_DS_SEARCH</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_DS_WRITE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_EXPLAIN_MAINT</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_INFO_HOST</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_INFO_SMS</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_INFO_SQL</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_INFO_SSID</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_INFO_SYSPARM</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_REXX</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_INFO_SYSPARM</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_NUMTCB1</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_JOB_CANCEL</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_PGM_CONTROL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_JOB_FETCH</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_PGM_CONTROL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_JOB_QUERY</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_PGM_CONTROL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_JOB_SUBMIT</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_PGM_CONTROL</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_TASK_ADD</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_TASK_CANCEL</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_TASK_LIST</td>
<td>DSNADM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_TASK_OUTPUT</td>
<td>DSNADM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Supplied routine</td>
<td>Schema name</td>
<td>Recommended value of NUMTCB</td>
<td>Core WLM environment</td>
<td>Does the routine have to run in an API-authorized environment?</td>
<td>Does the routine have to be program-controlled?</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>------------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>ADMIN_TASK_REMOVE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_TASK_STATUS</td>
<td>DSNADM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_TASK_UPDATE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_UPDATE_SYSPARM</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_REXX</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_UTIL_EXECUTE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_UTIL_MODIFY</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_UTIL_MONITOR</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_PGM_CONTROL</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADMIN_UTIL_SCHEDULE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADMIN_UTIL_SORT</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ALTER_JAVA_PATH</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CREATE_SESSION</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DB2_INSTALL_JAR</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DB2_REMOVE_JAR</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DB2_REPLACE_JAR</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DB2_UPDATE_JARINFO</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_ENDSESSIONMANAGER</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_INITIALIZECLIENT</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_LOOKUPSESSIONMANAGER</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_PINGSESSIONMANAGER</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_SENDCLIENTCOMMANDS</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_SENDCLIENTREQUESTS</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DBG_TERMINATECLIENT</td>
<td>SYSPROC</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DEBUGGERLEVEL</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DESTROY_SESSION</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSN_WLM_APPLENV</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNAACCOX</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNAJCICS</td>
<td>SYSPROC</td>
<td>40</td>
<td>DSNWLM_CICS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNAHVPVM</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNAIMS</td>
<td>SYSPROC</td>
<td>40</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNAIMS2</td>
<td>SYSPROC</td>
<td>40</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNTRBIND</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_REXX</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNLEUSR</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_NUMTCB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNTPSMP</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_REXX</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSNUTILS</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_UTILS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNUUTILU</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_UTILS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNUUTILV</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_UTILS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNWSPM</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_NUMTCB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSNWZP</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_NUMTCB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>GET_CONFIG</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GET_MESSAGE</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GET_REPORT</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GET_SYSTEM_INFO</td>
<td>SYSPROC</td>
<td>1</td>
<td>DSNWLM_NUMTCB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>INSTALL_JAR</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LIST_SESSION</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQREAD</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQREADALL</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Supplied routine</td>
<td>Schema name</td>
<td>Recommended value of NUMTCB</td>
<td>Core WLM environment</td>
<td>Does the routine have to run in an APF-authorized environment?</td>
<td>Does the routine have to be program-controlled?</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
<td>-----------------------------</td>
<td>----------------------</td>
<td>---------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>MQREADALLCLOB</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQREADCLOB</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQRECEIVE</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQRECEIVEALL</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQRECEIVEALLCLOB</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MQSEND</td>
<td>DB2MQ</td>
<td>≥ 10</td>
<td>DSNWLM_MQSERIES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PUT_COMMAND</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>QUERY_SESSION</td>
<td>DB2DEBUG</td>
<td>5 - 20</td>
<td>DSNWLM_DEBUGGER</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>REMOVE_JAR</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>REPLACE_JAR</td>
<td>SQLJ</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SET_PLAN_HINT</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SOAPHTTPNC</td>
<td>DB2XML</td>
<td>≥ 10</td>
<td>DSNWLM_WEBSERVICES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SOAPHTTPTPNC</td>
<td>DB2XML</td>
<td>≥ 10</td>
<td>DSNWLM_WEBSERVICES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SOAPHTTPPNV</td>
<td>DB2XML</td>
<td>≥ 10</td>
<td>DSNWLM_WEBSERVICES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SOAPHTTPTV</td>
<td>DB2XML</td>
<td>≥ 10</td>
<td>DSNWLM_WEBSERVICES</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQLCAMESSAGE</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQLCOL_PRIVILEGES</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQLCOLUMNS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQLFOREIGNKEYS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_FUNCTIONCOLS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQLFUNCTIONS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_GET_TYPE_INFO</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_PRIMARYKEYS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_PROCEDURECOLS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_PROCEDURES</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_PSEUDO_COLUMNS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_SPECIAL_COLUMNS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_STATISTICS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_TABLE_PRIVILEGES</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_TABLES</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SQL_UDTS</td>
<td>SYSIBM</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Alter</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Create</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Drop</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Encrypt</td>
<td>SYSFUN</td>
<td>2 - 8^2</td>
<td>DSNWLM_JAVA</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Restore</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Start</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Stop</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Takeover</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYSSTS.Update</td>
<td>SYSPROC</td>
<td>10 - 40</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>WLM.Refresh</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_GENERAL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>WLM_Set_Client_Info</td>
<td>SYSPROC</td>
<td>≥ 1</td>
<td>DSNWLM_GENERAL</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>XML_TRANSFORM</td>
<td>SYSFUN</td>
<td>2 - 8^2</td>
<td>DSNWLM_JAVA</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>XSR_ADD_SCHEMA_DOC</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_XML</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>XSR_Complete</td>
<td>SYSPROC</td>
<td>2 - 8^2</td>
<td>DSNWLM_JAVA</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>XSR_Register</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_XML</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>XSR_Remove</td>
<td>SYSPROC</td>
<td>40 - 60</td>
<td>DSNWLM_XML</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Supplied routine</td>
<td>Schema name</td>
<td>Recommended value of NUMTCB</td>
<td>Core WLM environment</td>
<td>Does the routine have to run in an APF-authorized environment?</td>
<td>Does the routine have to be program-controlled?</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>---------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
</tbody>
</table>

**Notes:**

1. This routine cannot run in an APF-authorized environment.

2. If you are running your Java stored procedures in 64-bit JVMs, the recommended NUMTCB is 25. You cannot run XSR_COMPLETE in a 64-bit Java virtual machine (JVM).

**Routine setup information in DSNTIDxx**

To install the Db2-supplied routines, job DSNTIJRT uses setup information that is stored in the CLIST defaults input member (DSNTIDxx). Each Db2-supplied routine is represented by a different name in DSNTIDxx. The following table shows the mapping between each DSNTIDxx name and Db2-supplied routine. The left column contains the DSNTIDxx name and the right column contains the routine name. The xxx in the DSNTIDxx names represent the letters that specify the WLM environment name, the grant execute list, and the package owner. For example, RTB01.xxx represents RTB01WLM, RTB01ACL, and RTB01PKO.

**Table 129. Mapping between DSNTIDxx variables and Db2-supplied routines**

<table>
<thead>
<tr>
<th>DSNTIDxx variable</th>
<th>Stored procedure name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTB01.xxx</td>
<td>DSNWSPM</td>
</tr>
<tr>
<td>RTB02.xxx</td>
<td>DSNWZP</td>
</tr>
<tr>
<td>RTCO1.xxx</td>
<td>DSNUTILS</td>
</tr>
<tr>
<td>RTCO2.xxx</td>
<td>DSNUTILU</td>
</tr>
<tr>
<td>RTCO3.xxx</td>
<td>DSNUTILV</td>
</tr>
<tr>
<td>RTD01.xxx</td>
<td>DSNACCOR</td>
</tr>
<tr>
<td>RTD02.xxx</td>
<td>DSNACCOX</td>
</tr>
<tr>
<td>RTE01.xxx</td>
<td>DSNLEUSR</td>
</tr>
<tr>
<td>RTF01.xxx</td>
<td>DSNTTBIND</td>
</tr>
<tr>
<td>RTF02.xxx</td>
<td>DSNTPSMP</td>
</tr>
<tr>
<td>RTF03.xxx</td>
<td>WLM_REFRESH</td>
</tr>
<tr>
<td>RTG01.xxx</td>
<td>ALTER_JAVA_PATH</td>
</tr>
<tr>
<td>RTG02.xxx</td>
<td>DB2_INSTALL_JAR</td>
</tr>
<tr>
<td>RTG03.xxx</td>
<td>DB2_REMOVE_JAR</td>
</tr>
<tr>
<td>RTG04.xxx</td>
<td>DB2_REPLACE_JAR</td>
</tr>
<tr>
<td>RTG05.xxx</td>
<td>DB2_UPDATEJARINFO</td>
</tr>
<tr>
<td>RTG06.xxx</td>
<td>INSTALL_JAR</td>
</tr>
<tr>
<td>RTG07.xxx</td>
<td>REMOVE_JAR</td>
</tr>
<tr>
<td>RTG08.xxx</td>
<td>REPLACE_JAR</td>
</tr>
<tr>
<td>RTH01.xxx</td>
<td>CREATE_SESSION</td>
</tr>
<tr>
<td>RTH02.xxx</td>
<td>DEBUGGERLEVEL</td>
</tr>
<tr>
<td>RTH03.xxx</td>
<td>DESTROY_SESSION</td>
</tr>
<tr>
<td>RTH04.xxx</td>
<td>GET_REPORT</td>
</tr>
<tr>
<td>RTH05.xxx</td>
<td>LIST_SESSION</td>
</tr>
</tbody>
</table>
Table 129. Mapping between DSNTIDxx variables and Db2-supplied routines (continued)

<table>
<thead>
<tr>
<th>DSNTIDxx variable</th>
<th>Stored procedure name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTH06xxx</td>
<td>PUT_COMMAND</td>
</tr>
<tr>
<td>RTH07xxx</td>
<td>QUERY_SESSION</td>
</tr>
<tr>
<td>RTH08xxx</td>
<td>DBG_ENDSESSIONMANAGER</td>
</tr>
<tr>
<td>RTH09xxx</td>
<td>DBG_INITIALIZECLIENT</td>
</tr>
<tr>
<td>RTH10xxx</td>
<td>DBG_LOOKUPSESSIONMANAGER</td>
</tr>
<tr>
<td>RTH11xxx</td>
<td>DBG_PINGSESSIONMANAGER</td>
</tr>
<tr>
<td>RTH12xxx</td>
<td>DBG_RECVCLIENTREPORTS</td>
</tr>
<tr>
<td>RTH13xxx</td>
<td>DBG_RUNSESSIONMANAGER</td>
</tr>
<tr>
<td>RTH14xxx</td>
<td>DBG_SENDCLIENTCOMMANDS</td>
</tr>
<tr>
<td>RTH15xxx</td>
<td>DBG_SENDCLIENTREQUESTS</td>
</tr>
<tr>
<td>RTH16xxx</td>
<td>DBG_TERMINATECLIENT</td>
</tr>
<tr>
<td>RTI03xxx</td>
<td>XSLTRANSFORM</td>
</tr>
<tr>
<td>RTI04xxx</td>
<td>XSR_ADDSCHEMADOC</td>
</tr>
<tr>
<td>RTI05xxx</td>
<td>XSR_COMPLETE</td>
</tr>
<tr>
<td>RTI06xxx</td>
<td>XSR_REGISTER</td>
</tr>
<tr>
<td>RTI07xxx</td>
<td>XSR_REMOVE</td>
</tr>
<tr>
<td>RTJ01xxx</td>
<td>SYSTS_CREATE</td>
</tr>
<tr>
<td>RTJ02xxx</td>
<td>SYSTS_DROP</td>
</tr>
<tr>
<td>RTJ03xxx</td>
<td>SYSTS_ENCRYPT</td>
</tr>
<tr>
<td>RTJ04xxx</td>
<td>SYSTS_RESTORE</td>
</tr>
<tr>
<td>RTJ05xxx</td>
<td>SYSTS_START</td>
</tr>
<tr>
<td>RTJ06xxx</td>
<td>SYSTS_STOP</td>
</tr>
<tr>
<td>RTJ07xxx</td>
<td>SYSTS_TAKEOVER</td>
</tr>
<tr>
<td>RTJ08xxx</td>
<td>SYSTS_UPDATE</td>
</tr>
<tr>
<td>RTJ09xxx</td>
<td>SYSTS_ALTER</td>
</tr>
<tr>
<td>RTK01xxx</td>
<td>SQLCAMESSAGE</td>
</tr>
<tr>
<td>RTK02xxx</td>
<td>SQLCOLPRIVILEGES</td>
</tr>
<tr>
<td>RTK03xxx</td>
<td>SQLCOLUMNS</td>
</tr>
<tr>
<td>RTK04xxx</td>
<td>SQLFOREIGNKEYS</td>
</tr>
<tr>
<td>RTK05xxx</td>
<td>SQLFUNCTIONCOLS</td>
</tr>
<tr>
<td>RTK06xxx</td>
<td>SQLFUNCTIONS</td>
</tr>
<tr>
<td>RTK07xxx</td>
<td>SQLGETTYPEINFO</td>
</tr>
<tr>
<td>RTK08xxx</td>
<td>SQLPRIMARYKEYS</td>
</tr>
<tr>
<td>RTK09xxx</td>
<td>SQLPROCEDURECOLS</td>
</tr>
<tr>
<td>RTK10xxx</td>
<td>SQLPROCEDURES</td>
</tr>
<tr>
<td>RTK11xxx</td>
<td>SQLSPECIALCOLUMNS</td>
</tr>
<tr>
<td>RTK12xxx</td>
<td>SQLSTATISTICS</td>
</tr>
<tr>
<td>RTK13xxx</td>
<td>SQLTABLEPRIVILEGES</td>
</tr>
<tr>
<td>RTK14xxx</td>
<td>SQLTABLES</td>
</tr>
<tr>
<td>RTK15xxx</td>
<td>SQLUDTS</td>
</tr>
<tr>
<td>DSNTIDxx variable</td>
<td>Stored procedure name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>RTL01xxx</td>
<td>ADMIN_COMMAND_DB2</td>
</tr>
<tr>
<td>RTL02xxx</td>
<td>ADMIN_COMMAND_DSN</td>
</tr>
<tr>
<td>RTL03xxx</td>
<td>ADMIN_COMMAND_UNIX</td>
</tr>
<tr>
<td>RTL04xxx</td>
<td>ADMIN_DS_BROWSE</td>
</tr>
<tr>
<td>RTL05xxx</td>
<td>ADMIN_DS_DELETE</td>
</tr>
<tr>
<td>RTL06xxx</td>
<td>ADMIN_DS_LIST</td>
</tr>
<tr>
<td>RTL07xxx</td>
<td>ADMIN_DS_RENAME</td>
</tr>
<tr>
<td>RTL08xxx</td>
<td>ADMIN_DS_SEARCH</td>
</tr>
<tr>
<td>RTL09xxx</td>
<td>ADMIN_DS_WRITE</td>
</tr>
<tr>
<td>RTL10xxx</td>
<td>ADMIN_INFO_HOST</td>
</tr>
<tr>
<td>RTL11xxx</td>
<td>ADMIN_INFO_SSID</td>
</tr>
<tr>
<td>RTL12xxx</td>
<td>ADMIN_JOB_CANCEL</td>
</tr>
<tr>
<td>RTL13xxx</td>
<td>ADMIN_JOB_FETCH</td>
</tr>
<tr>
<td>RTL14xxx</td>
<td>ADMIN_JOB_QUERY</td>
</tr>
<tr>
<td>RTL15xxx</td>
<td>ADMIN_JOB_SUBMIT</td>
</tr>
<tr>
<td>RTL17xxx</td>
<td>ADMIN_TASK_ADD</td>
</tr>
<tr>
<td>RTL18xxx</td>
<td>ADMIN_TASK_LIST</td>
</tr>
<tr>
<td>RTL19xxx</td>
<td>ADMIN_TASK_REMOVE</td>
</tr>
<tr>
<td>RTL20xxx</td>
<td>ADMIN_TASK_STATUS</td>
</tr>
<tr>
<td>RTL21xxx</td>
<td>ADMIN_UTL_SCHEDULE</td>
</tr>
<tr>
<td>RTL22xxx</td>
<td>ADMIN_UTL_SORT</td>
</tr>
<tr>
<td>RTL23xxx</td>
<td>ADMIN_TASK_CANCEL</td>
</tr>
<tr>
<td>RTL24xxx</td>
<td>ADMIN_TASK_OUTPUT</td>
</tr>
<tr>
<td>RTL25xxx</td>
<td>GET_CONFIG</td>
</tr>
<tr>
<td>RTL26xxx</td>
<td>GET_MESSAGE</td>
</tr>
<tr>
<td>RTL27xxx</td>
<td>GET_SYSTEM_INFO</td>
</tr>
<tr>
<td>RTL28xxx</td>
<td>ADMIN_INFO_SYSPARM</td>
</tr>
<tr>
<td>RTL29xxx</td>
<td>WLM_SET_CLIENT_INFO</td>
</tr>
<tr>
<td>RTL30xxx</td>
<td>ADMIN_UTL_EXECUTE</td>
</tr>
<tr>
<td>RTL31xxx</td>
<td>ADMIN_UTL_MODIFY</td>
</tr>
<tr>
<td>RTL32xxx</td>
<td>ADMIN_UTL_MONITOR</td>
</tr>
<tr>
<td>RTL33xxx</td>
<td>DSN_WLM_APPLENV</td>
</tr>
<tr>
<td>RTL34xxx</td>
<td>ADMIN_TASK_UPDATE</td>
</tr>
<tr>
<td>RTL35xxx</td>
<td>ADMIN_INFO_SMS</td>
</tr>
<tr>
<td>RTL36xxx</td>
<td>ADMIN_INFO_SQL</td>
</tr>
<tr>
<td>RTL37xxx</td>
<td>ADMIN_INFO_SYSLOG</td>
</tr>
<tr>
<td>RTL38xxx</td>
<td>ADMIN_UPDATE_SYSPARM</td>
</tr>
<tr>
<td>RTL40xxx</td>
<td>SET_PLAN_HINT</td>
</tr>
<tr>
<td>RTL41xxx</td>
<td>ADMIN_COMMAND_MVS</td>
</tr>
<tr>
<td>RTL42xxx</td>
<td>ADMIN_EXPLAIN_MAINT</td>
</tr>
</tbody>
</table>
Table 129. Mapping between DSNTIDxx variables and Db2-supplied routines (continued)

<table>
<thead>
<tr>
<th>DSNTIDxx variable</th>
<th>Stored procedure name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTM05xxx</td>
<td>DSNAHVPM</td>
</tr>
<tr>
<td>RTN01xxx</td>
<td>DSNACICS</td>
</tr>
<tr>
<td>RTN02xxx</td>
<td>DSNAIMS</td>
</tr>
<tr>
<td>RTN03xxx</td>
<td>DSNAIMS2</td>
</tr>
<tr>
<td>RTO02xxx</td>
<td>MQREAD</td>
</tr>
<tr>
<td>RTO03xxx</td>
<td>MQREADALL</td>
</tr>
<tr>
<td>RTO04xxx</td>
<td>MQREADALLCLOB</td>
</tr>
<tr>
<td>RTO05xxx</td>
<td>MQREADCLOB</td>
</tr>
<tr>
<td>RTO06xxx</td>
<td>MQRECEIVE</td>
</tr>
<tr>
<td>RTO07xxx</td>
<td>MQRECEIVEALL</td>
</tr>
<tr>
<td>RTO08xxx</td>
<td>MQRECEIVEALLCLOB</td>
</tr>
<tr>
<td>RTO09xxx</td>
<td>MQRECEIVECLOB</td>
</tr>
<tr>
<td>RTO10xxx</td>
<td>MQSEND</td>
</tr>
<tr>
<td>RTP01xxx</td>
<td>SOAPHTTPC</td>
</tr>
<tr>
<td>RTP02xxx</td>
<td>SOAPHTTPNC</td>
</tr>
<tr>
<td>RTP03xxx</td>
<td>SOAPHTTPNV</td>
</tr>
<tr>
<td>RTP04xxx</td>
<td>SOAPHTTPV</td>
</tr>
</tbody>
</table>

Related tasks:
- Migration step 24: Set up Db2-supplied routines
- Installing Db2-supplied routines during installation

Related reference:
- Core WLM environments for Db2-supplied routines

WLM management of stored procedures

Workload Manager (WLM) is a component of z/OS that manages incoming work requests for the operating system and allocates available system resources to meet these requests. Db2 for z/OS uses WLM to allocate workload requests for Db2 stored procedures.

When Db2 receives a request to run a stored procedure, WLM determines whether additional resources are needed to process the request. For example, WLM might determine that a new WLM-managed address space is needed. WLM makes this determination by evaluating whether the work for a service class is meeting the goals that you set. WLM also evaluates whether the work queues are becoming too large. WLM then examines the system resource utilization as part of the algorithm to decide if another address space should be started. In many cases, a sudden increase in incoming work is handled by existing WLM address spaces, and task control blocks (TCBs) remove work from the queues. Consequently, an address space can handle more concurrent stored procedures or user-defined functions than the number that is specified by NUMTCB in the application environment definition.

Requirement: All Db2 stored procedures (except native SQL procedures) must run in WLM-managed address spaces.
A WLM application environment starts and manages WLM address spaces that are to be used by stored procedures. You can either create your own WLM application environments or use existing ones, and each stored procedure must be assigned to a WLM application environment. You can assign more than one stored procedure to each WLM application environment, but you might want to separate them for various resource management reasons. The stored procedures that are assigned to a WLM application environment do not need to be written in the same high-level language. However, for different performance reasons, you might decide to separate stored procedures into different application environments by language.

You define application environments for stored procedures as part of your WLM service policy. You also need to define a JCL procedure that WLM is to use to start the address spaces. You store this procedure in PROCLIB. The application environment definition in your policy contains the name of this startup procedure in PROCLIB. If too few address spaces are available to meet the policy goal, WLM starts another server address space if the service policy definition allows it.

Related concepts:
- MVS Planning: Workload Management

Related tasks:
- Setting up a WLM application environment for stored procedures during installation
- Setting up a WLM application environment for stored procedures during migration

Related information:
- Setting the priority of stored procedures (Db2 Administration Guide)

Core WLM environments for Db2-supplied routines

Definitions for core WLM environments are provided as recommended initial environments for running Db2-supplied routines. You can use Db2 core WLM environments to model additional WLM environments as you tune your subsystem for maximum performance of the Db2-supplied routines that you use.

Use job DSNTIJRW to define each of these environments with the parameters below, including any changes that are applicable to your site. Alternatively, you can use the provided IWMARIN0 ISPF tool to define these environments. These Db2 core WLM environments are intended as a starting point, not as the optimal environments in which to run the Db2-supplied routines. Over time, you will identify routines for your site that function best in a separate WLM environment.

Installation job DSNTIJMV creates an address space procedure for each of the core WLM environments. The WLM environment address space names have the form ssnmWLMsuffix, where ssnm is the Db2 subsystem name specified on panel DSNTIPM in the SUBSYSTEM NAME field, and suffix is a character that differentiates the various WLM environments. Running DSNTIJMV adds these procedures to the system PROCLIB that you specified on panel DSNTIPW in the SYSTEM PROCEDURES field.

In the description of each environment below, the following terms are used:

**NUMTCB**
- The maximum number of procedures or functions that can run concurrently in the WLM-established stored procedure address space. The
NUMTCB setting for the environment is the maximum appropriate setting for at least one of the routines that should be run in that environment. Other routines that run in the environment may tolerate a higher NUMTCB setting.

APF-authorized
Whether the external module for one or more routines that are recommended to run in the environment must run under control of the z/OS authorized program facility (APF). “Yes” means that all libraries allocated by the STEPLIB DD must be APF-authorized. “No” means that at least one library allocated by the STEPLIB DD is not APF-authorized. Not all routines that are recommended to run in an APF-authorized environment necessarily require APF authorization.

ADMIN_UPDATE_SYSPARM and DSNTIPSMP must run in an environment that is not APF-authorized.

Special DDs
Any special data sets, libraries, or work files that are required by at least one of the routines that are recommended to run in the environment. These are in addition to the standard requirement of STEPLIB DD.

WLM environments

DSNWLM_DEBUGGER
This environment is appropriate for stored procedures that are provided for the Unified Debugger tool. Job DSNTIJMV creates an address space procedure for DSNWLM_DEBUGGER that is called DSNWLMD. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMD.

DSNWLM_DEBUGGER has the following characteristics:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMTCB</td>
<td>10</td>
</tr>
<tr>
<td>APF-authorized</td>
<td>No</td>
</tr>
<tr>
<td>Special DDs</td>
<td>This DD is optional:</td>
</tr>
<tr>
<td></td>
<td>• PSMDEBUG which is commented out in normal use. Uncomment it to activate debug tracing.</td>
</tr>
<tr>
<td>Other considerations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• This environment might have a high refresh rate when debug diagnostics are activated.</td>
</tr>
<tr>
<td></td>
<td>• The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.</td>
</tr>
</tbody>
</table>

DSNWLM_DSNACICS
This environment is appropriate for DSNACICS, the Db2 CICS transaction invocation stored procedure. Job DSNTIJMV creates an address space procedure for DSNWLM_DSACICS that is called DSNWLMC. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMC.

DSNWLM_DSACICS has the following characteristics:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMTCB</td>
<td>40</td>
</tr>
<tr>
<td>APF-authorized</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Special DDs: This DD is optional:

- DSNDUMP, which is commented out in normal use. Uncomment it to cause DSNACICS to generate an SVC dump when DSNACICS will generate an error message. When this DD is used, allocate it to output class for system dumps.

Other considerations:
- The CICS SDFSHEXCI library must be allocated by the STEPLIB DD.
- The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

**DSNWLM_GENERAL**

This environment is appropriate for most Db2-supplied routines. Job DSNTIJMV creates an address space procedure for DSNWLM_GENERAL that is called DSNWLMG. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMG.

DSNWLM_GENERAL has the following characteristics:

| NUMTCB | 40 |
| APF-authorized | Yes |

**DSNWLM_JAVA**

This environment is appropriate for most Db2-supplied Java routines. Job DSNTIJMV creates an address space procedure for DSNWLM_JAVA that is called DSNWLMJ. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMJ.

DSNWLM_JAVA has the following characteristics:

| NUMTCB | 5 |
| APF-authorized | No |

Special DDs: These DDs are required:

- JAVAENV allocates a data set that contains Language Environment runtime options for Java stored procedures. The presence of this DD statement indicates to Db2 that the WLM environment is for Java routines. For an interpreted Java routine, this data set must contain the environment variable JAVA_HOME. This environment variable indicates to Db2 that the WLM environment is for interpreted Java routines. JAVA_HOME also specifies the highest-level directory in the set of directories that contain the Java SDK.
- JSPDEBUG allocates a data set in which Language Environment puts runtime diagnostics.

Example:

```
//JAVAENV DD DISP=SHR,DSN=WLMJAVA.JPAENV
//JSPDEBUG DD SYSOUT=*  
```
Other considerations: By default, the DSNWLM startup procedure for DSNWLM_JAVA runs program DSNX9WLM, which is the WLM address space initialization program for a 31-bit JVM. If your JAVA_HOME variable specifies a path to a 64-bit JVM, such as /usr/lpp/java/8.0_64, you need to change DSNX9WLM in the EXEC statement to DSNX9WJM.

Related information:

WLM address space startup procedure for Java routines (Db2 Application Programming for Java)

The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside. If you are running your Java stored procedures in 64-bit JVMs, the recommended NUMTCB is 25.

DSNWLM_MQSERIES

This environment is appropriate for running Db2 MQSeries functions. Job DSNTIJMV creates an address space procedure for DSNWLM_MQSERIES that is called DSNWLMMM. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMM.

DSNWLM_MQSERIES has the following characteristics:

<table>
<thead>
<tr>
<th>NUMTCB</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>APF-authorized</td>
<td>No</td>
</tr>
<tr>
<td>Special DDs:</td>
<td>None</td>
</tr>
</tbody>
</table>
| Other         |MQSeries run time libraries must be allocated by the STEPLIB DD if they are not included in the system link list. Example:
 //STEPLIB DD DISP=SHR,DSN=prefix.RUNLIB.LOAD
 // DD DISP=SHR,DSN=prefix.SCEERUN
 // DD DISP=SHR,DSN=prefix.SDSNEXIT
 // DD DISP=SHR,DSN=prefix.SDSNLLOAD
 //*/ Uncomment the following if not included in the LINKLIST
 //*/ DD DISP=SHR,DSN=prefix.SCSQLOAD
 //*/ DD DISP=SHR,DSN=prefix.SCSQAUTH
 //*/ DD DISP=SHR,DSN=prefix.SCSQANL
| Other considerations: |MQSeries run time libraries must be allocated by the STEPLIB DD if they are not included in the system link list. Example:
 //STEPLIB DD DISP=SHR,DSN=prefix.RUNLIB.LOAD
 // DD DISP=SHR,DSN=prefix.SCEERUN
 // DD DISP=SHR,DSN=prefix.SDSNEXIT
 // DD DISP=SHR,DSN=prefix.SDSNLLOAD
 //*/ Uncomment the following if not included in the LINKLIST
 //*/ DD DISP=SHR,DSN=prefix.SCSQLOAD
 //*/ DD DISP=SHR,DSN=prefix.SCSQAUTH
 //*/ DD DISP=SHR,DSN=prefix.SCSQANL
| The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

DSNWLM_NUMTCB1

This environment is for Db2-supplied routines that must run serially. If a routine must run serially, it cannot share a task control block (TCB) with another routine. Routines that should run in this environment require a maximum TCB setting of 1 but have no other special DD requirements. Job DSNTIJMV creates an address space procedure for DSNWLM_NUMTCB1 that is called DSNWLMM. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMM.

DSNWLM_NUMTCB1 has the following characteristics:

<table>
<thead>
<tr>
<th>NUMTCB</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>APF-authorized</td>
<td>Yes</td>
</tr>
<tr>
<td>Special DDs:</td>
<td>None</td>
</tr>
</tbody>
</table>
Other considerations: The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

**DSNWLM_PGM_CONTROL**

This environment is appropriate for Db2-supplied routines that are registered to RACF program control or an equivalent and can therefore be loaded only with other similar routines. Job DSNTIJMV creates an address space procedure for DSNWLM_PGM_CONTROL that is called DSNWLMP. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMP.

DSNWLM_PGM_CONTROL has the following characteristics:

- **NUMTCB**: 40
- **APF-authorized**: Yes
- **Special DDs**: None
- **Other considerations**: The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

**DSNWLM_REXX**

This environment is primarily appropriate for the ADMIN_UPDATE_SYSPARM and DSNTPSMP stored procedures. However, it can be shared with the other Db2-supplied REXX stored procedures: DSNTBIND, ADMIN_COMMAND_DSN, and ADMIN_INFO_SYSLOG. Job DSNTIJMV creates an address space procedure for DSNWLM_REXX that is called DSNWLMR. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMR.

DSNWLM_REXX has the following characteristics:

- **NUMTCB**: 1
- **APF-authorized**: No
Special DDs: These DDs are required:

- SYSEXEC allocates the `prefix.SDSNCLST` library. This data set is where the Db2-supplied REXX routines reside.
- SYSTSPRT allocates an output data destination for messages from the TSO/E command processor.
- SQLDBRM allocates the data set for DBRMs created by running the DSNTPSMP stored procedure, and for DBRMs of packages that are bound by running the DSNTPSMP and DSNTBIND stored procedures.
- SQLCSRC allocates the data set for storing generated C code created by running DSNTPSMP.
- SQLLMOD allocates the data set for storing load modules created by running DSNTPSMP.
- SQLLIBC allocates C header files used by DSNTPSMP for compiling SQL procedures.
- SQLLIBL allocates include files used by DSNTPSMP for link editing external modules for SQL procedures.
- SYMSMSGS allocates the message file for the IBM Language Environment pre-link editor.
- CFGTPSMP (optional) allocates a site-provided sequential data set or member, used to define customized operation of DSNTPSMP in this application environment.
- SQLSRC allocates a work file that is used by DSNTPSMP. This work file needs to have a fixed block record format, a logical record length of 80, and a block size of 23440.
- SQLPRINT allocates a work file that is used by DSNTPSMP. This work file needs to have a variable block record format, a logical record length of 137, and a block size of 23476.
- SQLTERM allocates a work file that is used by DSNTPSMP. This work file needs to have a variable block record format, a logical record length of 137, and a block size of 23476.
- SQLOUT allocates a work file that is used by DSNTPSMP. This work file needs to have a variable block record format, a logical record length of 137, and a block size of 23476.
- SQLCPRT allocates a work file that is used by DSNTPSMP. This work file needs to have a variable block record format, a logical record length of 137, and a block size of 23476.
- SQLUT1 allocates a work file that is used by DSNTPSMP. This work file needs to have a fixed block record format, a logical record length of 80, and a block size of 23440.
- SQLUT2 allocates a work file that is used by DSNTPSMP. This work file needs to have a fixed block record format, a logical record length of 80, and a block size of 23440.
- SQLCIN allocates a work file that is used by DSNTPSMP. Do not specify a DCB for allocating this work file.
- SQLIN allocates a work file that is used by DSNTPSMP. This work file needs to have a fixed block record format, a logical record length of 80, and a block size of 3200.
- SYSMOD allocates a work file that is used by DSNTPSMP. This work file needs to have a fixed block record format, a logical record length of 80, and a block size of 23440.
- SQLDUMMY is required, but must be allocated to DUMMY.
- ZPMDFLTS allocates a library member that contains the names of data sets that are used by ADMIN_UPDATE_SYSPARM. ADMIN_UPDATE_SYSPARM dynamically allocates and deallocates these data sets.
Other considerations:

- The TSO/E DYNAMNBR parameter must indicate the number of system resources the system should hold for possible reuse.
- The STEPLIB DD must allocate at least one library that is not APF-authorized.
- The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

**DSNWLM_UTILS**

This environment is appropriate for Db2 utilities stored procedures, which require special DD allocations. These routines require a maximum NUMTCB setting of 1. Job DSNTIJMV creates an address space procedure for DSNWLM_UTILS that is called DSNWLMU. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMU.

DSNWLM_UTILS has the following characteristics:

<table>
<thead>
<tr>
<th>NUMTCB</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>APF-authorized</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Special DDs: These DDs are required:

- **UTPRINT** allocates a data set for messages from the sort program.
- **RNPRIN01** allocates a data set for messages from the sort program. This DD is required only if you plan to invoke RUNSTATS and collect distribution statistics.
- **STPRIN01** allocates a data set for messages from the sort program. This DD is required when a utility is directed to collect frequency statistics on at least one data-partitioned secondary index or when TABLESPACE TABLE COLGROUP FREQVAL is specified.
- **DSSPRINT** allocates a data set for messages when making concurrent copies.
- **SYSIN** allocates a work file for temporarily storing utility input statements.
- **SYSPRINT** allocates a work file for temporarily storing utility output messages.

Example:

```
//UTPRINT DD SYSOUT=*  
//RNPRIN01 DD SYSOUT=*  
//STPRIN01 DD SYSOUT=*  
//DSSPRINT DD SYSOUT=*  
//SYSIN DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SYSPRINT DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
```

Other considerations: The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

---

**DSNWLM_WEBSERVICES**

This environment is appropriate for running Db2 web services and SOAP functions. Job DSNTIJMV creates an address space procedure for DSNWLM_WEBSERVICES that is called DSNWLMW. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMW.

DSNWLM_WEBSERVICES has the following characteristics:
NUMTCB: 10
APF-authorized: No
Special DDs:
- This DD is required:
  - WSERROR allocates an HFS file for diagnostic output from the Db2 web services functions.

Example:
```
//WSERROR DD PATH='/tmp/wsc.err',
// PATHOPTS=(ORDWR,CREAT,APPEND),
// PATHMODE=(SRUSR,SIWUSR,SRGRP,)
// SRWGRP,SIROTH,SIWOTH)
```

Ensure that "PATH" specifies a valid path name and that the specified directory structure exists.

Other considerations:
- The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.
- The default timeout to read data from WS provider is 2 seconds. If you need to increase the timeout value, redefine the UDFs with the following run time options:
  ```
  RUN OPTIONS 'POSIX(ON),XPLINK(ON),ENVAR("DB2SOAP_TIMEOUT=X")'
  ```
  where X is a number that specifies the timeout value in seconds.

**DSNWLM_XML**

This environment is appropriate for Db2-supplied XML processing routines that require large amounts of memory. Job DSNTIJMV creates an address space procedure for DSNWLM_XML that is called DSNWLMX. The name and library name of this procedure change according to the Db2 subsystem name that is specified on panel DSNTIPM. For example, if the subsystem name is VA1A, the procedure is named VA1AWLMX.

DSNWLM_XML has the following characteristics:

NUMTCB: 40
APF-authorized: Yes
Special DDs: None
Other considerations: The STEPLIB DD must include DSN=prefix.SDSNEXIT or other library where the authorization exit modules (DSN3@ATH and DSN3@SGN) reside.

**Related concepts:**
Configuration control statements for Db2-supplied routines

**Related tasks:**
- Setting up a WLM application environment for stored procedures during installation
- Setting up a WLM application environment for stored procedures during migration

**DSNTWLMB tool**

The DSNTWLMB tool provides a batch interface for defining, installing, and activating WLM environments. You can also use DSNTWLMB to remove WLM application environments. This tool is invoked by job DSNTIJRW to create the core Db2 WLM environments.
Authorization

If the RACF facility class is active and a profile has been defined for the MVSADMIN.WLM.POLICY facility class, the user of this tool needs the following access:

- ACCESS(READ) for action WLMINFO
  PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY) ID(user) ACCESS(READ)
- ACCESS(UPDATE) for action ADD, ADD_ACTIVATE, ACTIVATE
  PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY) ID(user) ACCESS(UPDATE)

Parameter descriptions

The batch program accepts the following parameters:

**ACTION(action-type)**

Identifies the type of action that is to be performed. This parameter is required and it supports the following values:

**ACTIVATE**

The batch program activates a WLM service policy.

**ADD**

The batch program installs a new WLM application environment to an existing WLM service definition without activating a service policy. The new WLM application environment becomes available on the next activation of a WLM policy.

**ADD_ACTIVATE**

The batch program installs a new WLM application environment to an existing WLM service definition and activates a WLM service policy to enable the new WLM application environment.

**REMOVE**

The batch program removes a WLM application environment from an existing WLM service definition, and installs the modified service definition without activating it.

**REMOVE_ACTIVATE**

The batch program removes a WLM application environment from an existing WLM service definition, installs the modified service definition, and activates it.

**REMOVE_IF_EXISTS**

The same as REMOVE, except that if the WLM application environment to be removed does not exist, the return code is 4 (warning), instead of 8 (error).

**REMIFEXISTSACTIVATE**

The same as REMOVE_ACTIVATE, except that if the WLM application environment to be removed does not exist, the return code is 4 (warning), instead of 8 (error).

**WLMINFO**

The batch program queries the existing WLM service definition and returns basic information.

**POLICY(policy-ID)**

Identifies the WLM service policy that is to be activated. This parameter is required if any of the following actions are specified:

- ACTIVATE
ADD ACTIVATE
REMOVE ACTIVATE
REMIIFEXISTS ACTIVATE

The service policy ID can be 1 – 8 characters or “ACTIVE”. If “ACTIVE” is specified as the policy ID, the current active service policy is used.

SYSIN parameters

If the specified action is ADD or ADD_ACTIVATE, additional input is required from SYSIN to provide details about the WLM environment that is to be installed. The additional information can be entered from the SYSIN DD statement. The parameters can be entered on one line (up to 72 characters per line) or multiple lines, and they should be separated by one or more spaces.

If action-type is REMOVE, REMOVE_ACTIVATE, REMOVE_IF_EXISTS, or REMIFEXISTS_ACTIVATE, the WLMNAME parameter is required. Other SYSIN parameters are not valid with these action types.

The following parameters are supported under SYSIN DD:

WLMNAME(name)
The defined name for an application environment. The name can be 1 – 32 characters and cannot begin with “SYS”. This is a required parameter.

DESCRIPTION(description)
A 32 character area that describes the application environment. This parameter can be set to an empty value. For example:
DESCRIPTION()

PROCNAME(procedure-name)
Defines the JCL procedure that WLM uses to start server address spaces for the application environment. This is a required parameter.

STARTPARM(start-up-parameters)
Contains the parameters that WLM uses to start the JCL procedure. The parameters can be up to 115 characters, or it can be set to an empty value. If the parameters include the subsystem name, you can use the symbol &IWMSSNM to cause WLM to substitute the subsystem name instead of directly typing the subsystem name. This is useful because multiple instances of the subsystem with different names can use the application environment. For example:
STARTPARM(DB2SSN=&IWMSSNM,APPLENV=WLMENV1,NUMTCB=1)

WLMOPT(WLM_MANAGED|SINGLE_SERVER)
Tells WLM whether it should limit the number of server address spaces that are created. For a single system, the WLM_MANAGED option means that any number of servers can be created for the subsystem on the system where it is connected to Workload Manager. In a sysplex, WLM_MANAGED means that servers can be created for the subsystem on any number of systems in the sysplex. If multiple Db2 subsystems are on a system, WLM will create one server for each Db2 subsystem that calls the routines.

The SINGLE_SERVER option means that only one address space can exist per system. For Db2-supplied routines, you can set a limit of one address space per system if this limit is required by the routine. For example, if a server address space requires exclusive use of a resource, only a single address space can exist. Use this option if a stored procedure cannot execute concurrently in multiple address spaces. This option should be used if a stored procedure is to
be run in debug mode and writes to a trace. A limit of one address space per sysplex does not apply to Db2-supplied routines.

This parameter can be set to an empty value and will default to WLM_MANAGED.

Related tasks:
- Setting up a WLM application environment for stored procedures during migration
- Setting up a WLM application environment for stored procedures during installation

**Sample SQL statements for installing Db2-supplied routines**

Sample member DSNTESR contains the data definition language that is used by job DSNTIJRT.

Run job DSNTIJRT to create Db2-supplied routines. Sample member DSNTESR contains a full set of the data definition language that is processed by DSNTIJRT. DSNTESR contains statements to create and grant access to Db2-supplied routines, related objects, and the commands for binding packages.

**Configuration control statements for Db2-supplied routines**

Configuration control statements supply the basic input for the DSNTIRIN program, which installs and configures all routines that Db2 supplies.

Configuration control statements consist of a routine's schema and name (for stored procedures) or specific name (for user-defined functions), followed by customizable parameters. You can manually customize these parameters, but to save your settings and facilitate future migrations, specify changes using the Db2 routines: Advanced configuration options installation panels, which are accessed from the DSNTIPR1 panel.

**Parameters**

The following parameters are used in configuration control statements:

**WLMENV**
- The name of the WLM environment for running the routine. The default environment depends on the routine.

**GRANTTO**
- The authorization IDs allowed to execute the routine and use any associated tables or created global temporary tables. Separate multiple IDs with commas. Do not use blanks. The default is PUBLIC, which allows everyone to execute the routine.

**PKGOWNER**
- The authorization ID to own the routine's package. The specified owner must have the privileges required to execute the SQL statements contained in the routine. If this parameter is specified for a routine that does not have a package, the parameter is ignored. If this optional parameter is not specified, the default package owner is the primary ID used to run DSNTIRIN.

**Statement format**

The general form of a configuration control statement is:
schema.routine-name
  WLMENV(wlm-environment-name)
  GRANTTO(auth-ID1,...,auth-IDn)
  PKGOWNER(package-owner)

Statement rules

The following rules apply to configuration control statements:

- Statements must reside in records having a fixed-block record format with a logical record length of 80 (same as JCL).
- Only columns 1 - 72 of each record are inspected. Columns 73 - 80 are ignored.
- Statement elements are delimited by one or more blank characters.
- Statement elements cannot contain embedded blanks.
- Statements are delimited by one or more blank lines.
- A statement can be continued on multiple input records. A statement element can be continued by breaking it at column 72 of one record and resuming it at column 1 of the next record.
- An asterisk in column 1 of a record indicates a comment line and the content is ignored.
- Comment lines cannot begin after column 1.
- Comments and statement elements cannot be on the same line.
- Comment lines can be embedded within a statement.
- The routine must be specified as the schema name followed by a period and the routine name.
- The WLMENV parameter is required and must directly follow the schema.routine name.
- The GRANTTO parameter is required and must directly follow the WLMENV parameter.
- The PKGOWNER parameter is optional. If specified, it must follow the GRANTTO parameter.

Examples

The following examples are valid configuration control statements:

* Configuration statements for the Utilities stored procedures follow:
  SYSPROC.DSNUTILS
  WLMENV(WLMENV_DSNUTIL)
  *
  * A comment line within the statement is legal
  *
  GRANTTO(PUBLIC)
  PKGOWNER(ALFA)

  * Notice the all-blank line above to delineate the two statements
  SYSPROC.DSNUTILS
    WLMENV(WLMENV_DSNUTIL)
  *
  * Here is an example of wrapping an element to the next line
    GRANTTO(ALFA,BRAVO,CHARLIE,DELTA,ECHO,FOXTROT,GOLF,HOTEL,INDIA,JIUETT
    E,KILO,LIMA)
  *
  * PKGOWNER is commented-out below so the OWNER parm will be excluded
  * from the BIND PACKAGE statement
  * PKGOWNER(ALFA)
The following examples are invalid configuration control statements that violate the statement rules:

* Configuration statement for the WLM refresh stored procedure follows:
  * - Error 1: The statement must start with schema.routine-name
    WLM_REFRESH
  * - Error 2: Comments can't start after column 1
    WLMENV(WLMENV_APFAUTH)  * This is not a valid comment
  * - Error 3: Comments and elements can't be in the same line
    WLMENV(WLMENV_APFAUTH)  * This is an invalid place to put a comment
  * - Error 4: GRANTTO argument contains blanks
    GRANTTO( ALFA, BRAVO, CHARLIE, DELTA, ECHO, FOXTROT )
  * - Error 5: PKGOWNER is present but has no argument
    PKGOWNER()
  * - Error 6: The statements are not separated by a blank line
    SYSPROC.DSNTPSMP
  * - Error 7: The statement does not have a WLMENV parm
  * - Error 8: The statement contains a blank line
    GRANTTO(HOTEL)

Related reference:
Core WLM environments for Db2-supplied routines

Disabling Db2-supplied routines for validation

Job DSNTIJRT installs all Db2-supplied routines on your subsystem. After installation or migration, you can disable any routines that are unnecessary or unusable so the validation program can exclude it from processing.

Procedure

To disable a Db2-supplied routine:

Assign DSNWLM_NOT_ENABLED as the WLM environment of the routine. The DSNTRVfy validation program does not validate routines that are defined with DSNWLM_NOT_ENABLED as the WLM environment name.

Example

For example, if CICS is not installed, assign DSNWLM_NOT_ENABLED as the WLM environment of SYSPROC.DSNACICS by using the following statement:

```
ALTER PROCEDURE SYSPROC.DSNACICS WLM ENVIRONMENT DSNWLM_NOT_ENABLED
```

Setting up support for external SQL procedures

If you plan to create external SQL procedures by using DSNTPSMP or IBM Data Studio, you need to complete some additional setup. Alternatively, you can write your own JCL to create external SQL procedures, in which case you do not need to perform this setup.
About this task

DSNTPSMP is a REXX stored procedure that you can use to create and prepare external SQL procedures. DSNTPSMP is also used by IBM Data Studio to create external SQL procedures. When DSNTPSMP is invoked, Db2 uses the DSNTPSMP REXX EXEC, which is in the prefix.SDSNCLST data set.

Procedure

To set up support for external SQL procedures:

1. If DSNTPSMP does not already exist, run job DSNTIJR to create this stored procedure.
2. Install Db2 REXX Language Support.
3. Set up a WLM environment in which to run DSNTPSMP and create a WLM startup procedure.

Recommendation: If you do not have a WLM environment defined to run DSNTPSMP, use the core WLM environment DSNWLM_REXX.

Related concepts:

Job DSNTIJRT

Related tasks:

Installation step 18: Create default storage group, define temporary work files, and bind Db2 REXX Language Support: DSNTIJTM
Setting up a WLM environment for stored procedures during installation
Setting up a WLM environment for stored procedures during migration

Creating an external SQL procedure (Db2 Application programming and SQL)

Related reference:

Core WLM environments for Db2-supplied routines

Additional steps for enabling the stored procedures and objects for XML schema support

Db2 for z/OS provides several stored procedures that you can call in your application programs to perform XML schema registration and removal.

Before you begin

Before you complete this step, set up WLM application environments for the stored procedures for XML support, and either install the Db2-supplied routines during migration or install the Db2-supplied routines during installation.

About this task

Db2 provides the following stored procedures for XML schema support:

- SYSPROC.XSR_REGISTER
- SYSPROC.XSR_ADDSCHEMADOC
- SYSPROC.XSR_COMPLETE
Db2 provides the following tables for XML schema support:

- SYSIBM.XSRANNOTATIONINFO
- SYSIBM.XSRCOMPONENT
- SYSIBM.XSOBJECTCOMPONENTS
- SYSIBM.XSOBJECTHIERARCHIES
- SYSIBM.XSOBJECTGRAMMAR
- SYSIBM.XSOBJECTPROPERTY
- SYSIBM.XSOBJECTS
- SYSIBM.XSRPROPERTY

**Procedure**

To enable the stored procedures and objects for XML schema support:

Install Java support. The XSR_COMPLETE stored procedure is a Java stored procedure and therefore requires Java support.

**Related concepts:**

- Support for Java
- Job DSNTIJRT

**Related reference:**

- Core WLM environments for Db2-supplied routines
- XML processing routines panel: DSNTIPRI

**Related information:**

- Procedures for XML schema registration and removal that are supplied with Db2 (Db2 Programming for XML)

### Additional steps for enabling the function for XSLTRANSFORM routines support

Db2 for z/OS provides a function that you can call in your application programs to transform XML documents with XSL transformation processing.

**Before you begin**

Before you complete this step, [set up WLM application environments](#) for Java routines, and either [install the Db2-supplied routines during migration](#) or [install the Db2-supplied routines during installation](#).

**About this task**

Db2 provides the following function for XSL transform support:

- SYSFUN.XSLTRANSFORM

**Procedure**

To enable the user-defined function for XSL transformation support:
Install Java support.
The XSLTRANSFORM function is a Java stored procedure and therefore requires Java support.

**Related concepts:**
- Support for Java
- Job DSNTIJRT

**Related reference:**
- Core WLM environments for Db2-supplied routines
- XML processing routines panel: DSNTIPRI

### Additional steps for enabling the administrative task scheduler and administrative enablement routines

The administrative task scheduler routines are Db2-supplied routines that enable you to schedule work. The administrative enablement routines are Db2-supplied routines that enable you to execute administrative work immediately.

#### Before you begin

Before you complete these steps, [set up WLM application environments](#) for the administrative task scheduler and administrative enablement routines, and either [install the Db2-supplied routines during migration](#) or [install the Db2-supplied routines during installation](#).

**Recommendation:** If you do not have a WLM environment defined to run these routines, use Db2 core WLM environment DSNWLM_GENERAL, except as follows:

- **ADMIN_COMMAND_DSN** - Use DSNWLM_REXX
- **ADMIN_COMMAND_UNIX** - Use DSNWLM_PGM_CONTROL
- **ADMIN_INFO_SYSLOG** - Use DSNWLM_REXX
- **ADMIN_INFO_SYSPARM** - Use DSNWLM_NUMTCB1
- **ADMIN_JOB_CANCEL** - Use DSNWLM_PGM_CONTROL
- **ADMIN_JOB_FETCH** - Use DSNWLM_PGM_CONTROL
- **ADMIN_JOB_QUERY** - Use DSNWLM_PGM_CONTROL
- **ADMIN_JOB_SUBMIT** - Use DSNWLM_PGM_CONTROL
- **ADMIN_UPDATE_SYSPARM** - Use DSNWLM_REXX
- **ADMIN_UTL_MONITOR** - Use DSNWLM_PGM_CONTROL

#### About this task

The following routines are administrative task scheduler routines:

- **ADMIN_TASK_ADD**
- **ADMIN_TASK_CANCEL**
- **ADMIN_TASK_LIST**
- **ADMIN_TASK_OUTPUT**
- **ADMIN_TASK_REMOVE**
- **ADMIN_TASK_STATUS**
- **ADMIN_TASK_UPDATE**

The following routines are administrative enablement routines:
Procedure

To enable the administrative task scheduler and administrative enablement routines:

1. In the JCL for starting the WLM-established address space for running the stored procedures whose load module must reside in an APF-authorized library, ensure that all libraries in the STEPLIB DD concatenation are APF-authorized.

2. Take the following steps, which are required because the started task module, DSNADMT0, of the administrative task scheduler switches users by using the pthread_security_np() function of z/OS UNIX System Services:
   a. Ensure that the started task user (STARTUID) has READ access to the BPX.SERVER FACILITY class profile by taking the action that is recommended in the following table. The action that you take depends on whether the BPX.SERVER is defined and whether the STARTUID is a superuser.
Table 130. Ensuring that the STARTUID has READ access to the BPX.SERVER FACILITY class profile

<table>
<thead>
<tr>
<th>Is the BPX.SERVER in FACILITY class defined?</th>
<th>Is the STARTUID a superuser?</th>
<th>Actions to take</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>1. Define the BPX.SERVER class profile in the FACILITY class.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Permit STARTUID READ access to this profile.</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No action required</td>
</tr>
<tr>
<td>Yes</td>
<td>n/a</td>
<td>Permit STARTUID READ access to resource profile BPX.SERVER in the FACILITY class.</td>
</tr>
</tbody>
</table>

b. Ensure that the scheduler program is program controlled by taking one of the following actions:
   • Define the scheduler program to RACF program control, by customizing and running job step DSNADSR in job DSNTIJRA.
   • Define the BPX.DAEMON.HFSCTL class profile in the FACILITY class, and permit STARTUID READ access to this profile.

3. If the BPX.DAEMON facility class is active and the BPX.DAEMON.HFSCTL facility class is not defined, perform the following actions for the stored procedures that must be registered to RACF program control. The stored procedures that require RACF program control are:
   • ADMIN_COMMAND_UNIX
   • ADMIN_JOB_CANCEL
   • ADMIN_JOB_FETCH
   • ADMIN_JOB_QUERY
   • ADMIN_JOB_SUBMIT

a. Customize and run job step DSNADER in job DSNTIJRA to define the security environment for these stored procedures. You need to uncomment job step DSNADER before you run job DSNTIJRA. You do not need to run any other job steps in job DSNTIJRA.

b. Ensure that you have a WLM environment for these stored procedures that meets the following requirements:
   • The WLM-established stored procedure address space loads only controlled programs.
   • In the JCL for starting the WLM-established address space for running these stored procedures, ensure that all libraries in the STEPLIB DD concatenation are APF-authorized.
   • The name of this environment must match the WLMENV parameter value in the configuration control statement for each stored procedure in job DSNTIJRT.

Related reference:

- Procedures that are supplied with Db2 (Db2 SQL)
- Core WLM environments for Db2-supplied routines
- Db2 administration routines panel: DSNTIPRL
Additional steps for enabling the CICS transaction invocation procedure

The CICS transaction invocation stored procedure, DSNACICS, invokes CICS server programs.

Before you begin

Before you complete these steps, either install the Db2-supplied routines during migration or install the Db2-supplied routines during installation.

Procedure

To enable the CICS transaction invocation procedure:

1. Optional: Set up a WLM application environment specifically for running DSNACICS.
   Although DSNACICS can run in a WLM address space with other stored procedures, DSNACICS performs better if it runs in its own address space.

   **Recommendation:** If you do not have a WLM environment defined to run this routine, use Db2 core WLM environment DSNWLM_CICS.

2. Create or modify the JCL startup procedure for the stored procedure address space for running DSNACICS.
   If you ran installation job DSNTIJMV, such a sample startup procedure has been created in your SYS1.PROCLIB data set. The generic name of the procedure is DSNWLMC. The CICS data set names in the procedure are based on the values that you specified on panel DSNTIPW. If needed, you can modify this startup procedure.
   If you did not run installation job DSNTIJMV or the part of it that creates this startup procedure, run the job now. You can also use the sample startup procedure in the job as a model to create your own startup procedure.

Related concepts:
- Job DSNTJRT

Related tasks:
- Assigning stored procedures and functions to WLM application environments (Db2 Performance)

Related reference:
- Core WLM environments for Db2-supplied routines
- DSNTIPW: Other data set names panel
- DSNACICS stored procedure (Db2 SQL)

Related information:

Additional steps for enabling the common SQL API stored procedures

You can use the common SQL API stored procedures to retrieve information about the data server configuration, the short message text for an SQL code, and system information. These stored procedures are used primarily by tools.
Before you begin

Before you complete these steps, either install the Db2-supplied routines during migration or install the Db2-supplied routines during installation. To create the GET_CONFIG stored procedure with the correct security attributes, job DSNTIJRT must be run by a user ID with SYSOPR and MONITOR1 privileges.

Job DSNTIJRT installs and configures these stored procedures, but you must complete additional steps before you use them.

If GET_SYSTEM_INFO will be called to return the ICF catalog information about Db2 data sets, the user who defines the stored procedure must have MONITOR1 and DISPLAY privileges. If GET_SYSTEM_INFO will be called to return catalog attributes or ICF catalog information about Db2 data sets, the user who defines the stored procedure must have READ access to the OPERCMDS resource MVS.MCSOPER.DSNADMGS. If MVS.MCSOPER.DSNADMGS does not exist, the user must have READ access to the OPERCMDS resource MVS.MCSOPER.*. Otherwise, catalog attributes information will not be returned from GET_SYSTEM_INFO. To give the user that defines the procedures access to the RACF OPERCMDS class, the RACF security administrator must:

1. If the OPERCMDS class is not active, issue the following SETROPTS command to activate the OPERCMDS class:
   ```
   SETROPTS CLASSACT(OPERCMDS)
   ```
2. If the generic profile for OPERCMDS is not active, issue the following SETROPTS command to activate it:
   ```
   SETROPTS GENERIC(OPERCMDS)
   ```
3. Issue the RDEFINE command to establish a profile for MVS.MCSOPER.console-name. In this example, `console-name=DSNADMGS`:
   ```
   RDEFINE OPERCMDS MVS.MCSOPER.DSNADMGS UACC(NONE)
   ```
4. Give the user ID access to the class. In this example, the user ID is USER001:
   ```
   PERMIT MVS.MCSOPER.DSNADMGS CLASS(OPERCMDS) ID(USER001) ACCESS(READ)
   ```
5. Issue the SETROPTS RACLIST command to refresh the OPERCMDS reserve class:
   ```
   SETROPTS RACLIST(OPERCMDS) REFRESH
   ```

Before you complete the additional steps to enable the common SQL API stored procedures, set up WLM application environments for them to run in. These stored procedures run in WLM-established stored procedures address spaces.

The common SQL API stored procedures are:

- `SYSPROC.GET_CONFIG`
- `SYSPROC.GET_MESSAGE`
- `SYSPROC.GET_SYSTEM_INFO`
- `SYSPROC.SET_PLAN_HINT`

GET_SYSTEM_INFO must run in a separate WLM environment from DSNUTILS or DSNUTILU.

Procedure

To enable the common SQL API stored procedures:
In the JCL for starting the WLM-established address space for running the GET_SYSTEM_INFO stored procedure, ensure that all libraries in the STEPLIB DD concatenation are APF-authorized.

Related concepts:
- Common SQL API stored procedures (Db2 Administration Guide)

Related reference:
- GET_CONFIG stored procedure (Db2 SQL)
- GET_MESSAGE stored procedure (Db2 SQL)
- GET_SYSTEM_INFO stored procedure (Db2 SQL)
- SET_PLAN_HINT stored procedure (Db2 SQL)

**Additional steps for enabling WebSphere MQ user-defined functions**

Before applications can call the WebSphere MQ user-defined functions, you must enable the user-defined functions.

**Before you begin**

If you want to use the MQI-based MQ functions, you must install WebSphere MQ Version 6 or later.

**About this task**

Related reference:
- Systems Programmer's Guide to: Workload Manager (IBM Redbooks)
- WebSphere MQ library

**Moving from previous versions of the WebSphere MQ user-defined functions**

During installation or migration, the current versions of some WebSphere MQ user-defined functions are installed under schema DB2MQ by job DSNTIJRT.

**About this task**

The following user-defined functions have new versions:
- MQREAD
- MQREADALL
- MQREADALLCLOB
- MQREADCLOB
- MQRECEIVE
- MQRECEIVEALL
- MQRECEIVEALLCLOB
- MQRECEIVECLOB
- MQSEND
Recommendation: Code new applications to use the functions in schema DB2MQ. The functions in schemas DB2MQ1C, DB2MQ2C, DB2MQ1N, and DB2MQ2N are no longer supported.

Converting from the AMI-based MQ functions to the MQI-based MQ functions
The MQI-based MQ functions use two tables, SYSIBM.MQSERVICE_TABLE and SYSIBM.MQPOLICY_TABLE, instead of Application Messaging Interface (AMI) configuration files. You need to bind the new functions, create the tables, and convert the contents of the AMI configuration files to rows in the MQ tables.

About this task
Job DSNTIJRT binds the new functions and creates the tables. You need to manually convert the contents of the AMI configuration files.

Procedure
To manually move the AMI configuration file data to the MQ tables:
1. Map sub-elements and attributes under the <ServicePoint> element to columns in SYSIBM.MQSERVICE_TABLE as indicated in the following table.

<table>
<thead>
<tr>
<th>Table 131. How elements and attributes under the &lt;ServicePoint&gt; element map to columns in the SYSIBM.MQSERVICE_TABLE table</th>
<th>Element</th>
<th>Attribute of element</th>
<th>Column</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ServicePoint&gt;</td>
<td>longDescription</td>
<td>DESC_LONG</td>
<td>&lt;ServicePoint longDescription=&quot;&quot; shortDescription=&quot;&quot; xmi.label=&quot;DB2.DEFAULT.SERVICE&quot;/&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shortDescription</td>
<td>DESC_SHORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xmi.label</td>
<td>SERVICENAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;ServicePoint_queueName&gt;</td>
<td>value</td>
<td>INPUTQUEUE</td>
<td>&lt;ServicePoint_queueName .. value=&quot;DB2MQ_DEFAULT_Q&quot;/&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;ServicePoint_queueMgrName&gt;</td>
<td>value</td>
<td>QUEUEMANAGER</td>
<td>&lt;ServicePoint_queueMgrName .. value=&quot;MQND&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>

2. Map sub-elements and attributes under the <Policy> element to columns in SYSIBM.MQPOLICY_TABLE as indicated in the following table.

<table>
<thead>
<tr>
<th>Table 132. How elements and attributes under the &lt;Policy&gt; element map to columns in the SYSIBM.MQPOLICY_TABLE table</th>
<th>Element</th>
<th>Attribute of element</th>
<th>Value of attribute</th>
<th>Column</th>
<th>Value of column</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Policy&gt;</td>
<td>longDescription</td>
<td>value</td>
<td>DESC_LONG</td>
<td>value</td>
<td>&lt;Policy longDescription=&quot;&quot; shortDescription=&quot;&quot; xmi.label=&quot;DB2.DEFAULT.POLICY&quot;/&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shortDescription</td>
<td>DESC_SHORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xmi.label</td>
<td>POLICYNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;GeneralAttributes_syncpoint&gt;</td>
<td>value</td>
<td>Yes</td>
<td>SYNCPOINT</td>
<td>Y</td>
<td>&lt;GeneralAttributes_syncpoint .. value=&quot;Yes&quot;/&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;SendAttributes_priority&gt;</td>
<td>value</td>
<td>AsTransport</td>
<td>SEND_PRIORITY</td>
<td>-1</td>
<td>&lt;SendAttributes_priority .. value=&quot;AsTransport&quot;/&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Table 132. How elements and attributes under the `<Policy>` element map to columns in the SYSIBM.MQPOLICY_TABLE table (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute of element</th>
<th>Value of attribute</th>
<th>Column</th>
<th>Value of column</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;SendAttributes persistence&gt;</code></td>
<td>value</td>
<td>AsTransport</td>
<td>SEND _PERSISTENCE</td>
<td>Q</td>
<td><code>&lt;SendAttributes persistence .. value=&quot;AsTransport&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes expiry&gt;</code></td>
<td>value</td>
<td>Unlimited</td>
<td>SEND_EXPIRY</td>
<td>-1</td>
<td><code>&lt;SendAttributes_expiry .. value=&quot;Unlimited&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SendAttributes retryCount&gt;</code></td>
<td>value</td>
<td>n</td>
<td>SEND_RETRY_COUNT</td>
<td>n</td>
<td><code>&lt;SendAttributes_retryCount .. value=&quot;0&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;SendAttributes retryInterval&gt;</code></td>
<td>value</td>
<td>n</td>
<td>SEND_RETRY_INTERVAL</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes newCorrelId&gt;</code></td>
<td>value</td>
<td>Yes</td>
<td>SEND_NEW_CORRELID</td>
<td>Y</td>
<td><code>&lt;SendAttributes_newCorrelId .. value=&quot;Yes&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes responseCorrelId&gt;</code></td>
<td>value</td>
<td>MessageID</td>
<td>SEND_RESPONSE_CORRELID</td>
<td>C</td>
<td><code>&lt;SendAttributes_responseCorrelId .. value=&quot;MessageID&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CorrelID</td>
<td></td>
<td>P</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes exceptionAction&gt;</code></td>
<td>value</td>
<td>DLQ</td>
<td>SEND_EXCEPTION_ACTION</td>
<td>Q</td>
<td><code>&lt;SendAttributes_exceptionAction .. value=&quot;DLQ&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Discard</td>
<td></td>
<td>D</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes reportType&gt;</code></td>
<td>value, value¹</td>
<td>No</td>
<td>SEND_REPORT_EXCEPTIONDESC_LONG</td>
<td>N</td>
<td><code>&lt;SendAttributes_reportType .. value=&quot;No&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td></td>
<td>E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Data</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Full_Data</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes reportTypeCOA&gt;</code></td>
<td>value, value¹</td>
<td>No</td>
<td>SEND_REPORT_COA</td>
<td>N</td>
<td><code>&lt;SendAttributes_reportTypeCOA .. value=&quot;No&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td></td>
<td>C</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Data</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Full_Data</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes reportTypeCOD&gt;</code></td>
<td>value, value¹</td>
<td>No</td>
<td>SEND_REPORT_COD</td>
<td>N</td>
<td><code>&lt;SendAttributes_reportTypeCOD .. value=&quot;No&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td></td>
<td>C</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Data</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Full_Data</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td><code>&lt;SendAttributes reportTypeExpiration&gt;</code></td>
<td>value, value¹</td>
<td>No</td>
<td>SEND_REPORT_EXPIRY</td>
<td>N</td>
<td><code>&lt;SendAttributes_reportTypeExpiration .. value=&quot;No&quot;/&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td></td>
<td>E</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Data</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes, Report</td>
<td>_With_Full_Data</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

1. `value¹` indicates multiple values are possible for the attribute.
Table 132. How elements and attributes under the &lt;Policy&gt; element map to columns in the SYSIBM.MQPOLICY_TABLE table (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Value of attribute</th>
<th>Column</th>
<th>Value of column</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ReceiveAttributes _waitInterval&gt;</td>
<td>value</td>
<td>RCV_WAIT_INTERVAL</td>
<td>n</td>
<td>&lt;ReceiveAttributes _waitInterval .. value=&quot;60&quot;/&gt;</td>
</tr>
<tr>
<td>&lt;ReceiveAttributes _convert&gt;</td>
<td>value</td>
<td>RCV_CONVERT</td>
<td>Y</td>
<td>&lt;ReceiveAttributes _convert .. value=&quot;Yes&quot;/&gt;</td>
</tr>
<tr>
<td>&lt;ReceiveAttributes _acceptTruncated Messages&gt;</td>
<td>value</td>
<td>RCV_ACCEPT_TRUNC_MSG</td>
<td>Y</td>
<td>&lt;ReceiveAttributes _acceptTruncated Messages .. value=&quot;Yes&quot;/&gt;</td>
</tr>
<tr>
<td>&lt;ReceiveAttributes _openShared&gt;</td>
<td>value</td>
<td>RCV_OPEN_SHARED</td>
<td>S</td>
<td>&lt;ReceiveAttributes _openShared .. value=&quot;Yes&quot;/&gt;</td>
</tr>
</tbody>
</table>

Note:

1. The two values are used in conjunction to determine the column value.

Starting the queue manager

To enable WebSphere MQ user-defined functions, start the queue manager after you create and configure the broker domain.

Procedure

To start the queue manager:

1. Issue the following command from the z/OS console, where &lt;command-prefix-string&gt; is the command prefix string for the WebSphere MQ subsystem:
   &lt;command-prefix-string&gt; START QMGR
   For example, if the command prefix string for your WebSphere MQ subsystem is –MQND, issue this command:
   –MQND START QMGR

2. To check whether the queue manager is available, issue the following command from the TSO Command Processor panel, which is option 6 of the ISPF/PDF primary options menu:
   CSQOREXX

Customizing WLM application environments for running WebSphere MQ user-defined function support

To run WebSphere MQ user-defined functions, you need to set up two different WLM application environments and corresponding WLM startup procedures.

About this task

The WLM environments that you must set up are:

- An application environment and startup procedure for running the single-phase commit functions
- An application environment and startup procedure for running the two-phase commit functions

Related concepts:
Verifying the Db2 and WebSphere MQ setup

After you set up the WebSphere MQ environment, you can verify that it is set up correctly by running Db2 jobs.

Procedure

To verify your WebSphere MQ setup:

1. Customize and run job DSNTEJMQ to verify that your WebSphere MQ environment is set up correctly for invoking Db2 WebSphere MQ user-defined functions. Instructions for customizing this job are in the job prolog. Job DSNTEJMQ defines a local queue and invokes each of the WebSphere MQ functions through DSNTEP2.
2. Customize and run jobs DSNTEJSQ and DSNTEJSV to verify that your WebSphere MQ environment is set up correctly for invoking Db2 WebSphere MQ publish/subscribe user-defined functions. Instructions for customizing these jobs are in the job prolog.

Enabling Db2 web services

Web services are sets of business functions that applications or other web servers invoke over the Internet by using standard HTTP or HTTPS requests.

Additional steps for enabling web service user-defined functions

Enabling the web service user-defined functions allows you to specify a complete SOAP message as input, so you receive the complete SOAP messages from the web service provider's response.

Before you begin

Before you complete these steps, set up WLM application environments for web service user-defined functions.

Recommendation: For web service functions, use the DSNWLM_WEBSERVICES core WLM environment.

About this task

During installation or migration, the following web service user-defined functions were installed by job DSNTIJRT:

- DSNWSCVV
- DSNWSCVC
- DSNWSCCV
- DSNWSCCC

Procedure

To set up the web service user-defined functions:

1. The default timeout to read data from a web service provider is two seconds. Increase the timeout value by redefining the web service user-defined functions
with the key and value pair DB2SOAP_TIMEOUT=X in the ENVAR parameter values, where X is a number that specifies the timeout value in seconds.

2. If you want to use SOAP Version 1.2, include the key and value pair DB2SOAP_VERSION=SOAP_12 in the ENVAR parameter values. By default, Db2 uses SOAP Version 1.1. If you specify any other value for DB2SOAP_VERSION besides SOAP_12, SOAP Version 1.1 is assumed.
   If your subsystem uses SOAP Version 1.1 and SOAP Version 1.2, you must define two sets of SOAP user-defined functions: one without ENVAR("DB2SOAP_VERSION=SOAP_12"), and one with ENVAR("DB2SOAP_VERSION=SOAP_12").

3. If you want to receive SOAP error messages, specify the ENVAR option RETURN_ERROR=YES.
   If RETURN_ERROR=NO or the option is not specified and an error occurs, the web service provider returns a SOAP response to the client with an HTTP return code set other than 200. In this case, the user-defined functions end with SQLCODE=-443 and the complete HTTP response that was sent by web service provider is stored in the WSERROR DD file, as specified in the WLM startup procedure. The file contains the actual bytes that are sent by the web service provider.
   If you specify RETURN_ERROR=YES, the SOAPHTTP user-defined function writes the HTTP content to the WSERROR DD file when the HTTP status code is not 200. It also builds a new XML document with an error element node that contains an "httprc" attribute around the SOAP response, as follows:
   
   ```xml
   <error httprc="{HTTP status code}">
     {SOAP response}
   </error>
   ```

   The XML document is returned without issuing SQLCODE -443. Your applications must check for a root element node of error to determine if an HTTP error is returned. If an HTTP error is returned, the applications must process the SOAP response to handle the error.

4. If you do not use the DSNWLM_WEBSERVICES core WLM environment, add the following DD card to the WLM startup procedure where the user-defined functions are defined.

   ```
   // WSERROR DD PATH='<HFS filename>',
   // PATHOPTS=(ORDWR,OCREAT,OAPPEND),
   // PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP,SIROTH,SIWOTH)
   ```

   where `<HFS filename>` is the name of an HFS, such as /tmp/wsc.err.
   If the response is in Unicode, you can browse the file on an MVS system by issuing the following command:
   
   ```bash
   iconv -f UTF-8 -t IBM-1047 < tmp/wsc.err
   ```

   You can store the contents in EBCDIC by saving it to another file by issuing the following command:
   
   ```bash
   iconv -f UTF-8 -t IBM-1047 < wsc.txt > wsc_ebcdic.txt
   ```

   You can also FTP the file in BINARY mode to a workstation and open it using a text editor.

**Related concepts:**
- The SOAPHTTPNV and SOAPHTTPNC user-defined functions (Db2 Application programming and SQL)

**Related reference:**

Chapter 11. Working with additional capabilities for Db2
Supported web service calls:

The web service user-defined functions support unsecured web services, HTTP Basic Authorized web services, and HTTPS Secured web services.

1. Unsecured web services
   For unsecured web services, the HTTP request that contains SOAP is sent in clear text to the web service provider, where no authentication is performed. The endpoint URL that is specified during invocation of the web service user-defined functions has the following form:
   \[http://[hostname[:port]]/[path]\]

2. HTTP Basic Authorized web services
   For HTTP Basic Authorized web services, the specified user and password is encoded in Base64 and sent as part of an HTTP Header. The HTTP request that contains SOAP is sent in clear text to the web service provider, where it authenticates the supplied user and password before processing the request. The endpoint URL that is specified during invocation of the web service user-defined functions has the following form:
   \[http://[user[:password]@[hostname[:port]]/[path]\]

3. HTTPS Secured web services
   For HTTPS Secured web services, as part of the HTTPS protocol, the complete HTTP request that contains SOAP is encrypted and sent to the web service provider. The endpoint URL that is specified during invocation of the web service user-defined functions has the following form:
   \[https://[hostname[:port]]/[path]\]

Enabling the web service user-defined function support for HTTPS:

Support for HTTPS in the web service user-defined functions uses Application Transparent Transport Layer Security (AT-TLS). AT-TLS is based on z/OS System SSL, and it transparently implements Transport Layer Security (TLS) protocols in the TCP layer of the stack.

About this task

SSL connections make use of public/private key mechanisms (PKI) for authenticating each side of the SSL session and agreeing on bulk encryption keys to be used for the SSL session. To use PKI, public/private key pairs must be generated. In addition, X.509 certificates (which contain public keys) might need to be created, or certificates must be requested, received, and managed.

System SSL supports the following two methods for managing PKI private keys and certificates:

- A z/OS shell-based program called gskkyman. gskkyman creates, fills in, and manages a z/OS file that contains PKI private keys, certificate requests, and certificates. This z/OS file is called a key database and, by convention, has a file extension of .kdb.
- The z/OS Security Server (RACF) RACDCERT command. RACDCERT installs and maintains PKI private keys and certificates in RACF.
Procedure

To enable support for HTTPS and AT-TLS:

1. Specify the TTLS parameter on the TCPCONFIG statement in PROFILE.

2. To protect TCP/IP connections, you can configure the RACF EZB.INITSTACK.sysname.tcpname resource in the SERVAUTH class to block all stack access except for the user IDs that are permitted to use the resource. Refer to member EZARACF in sample data set hlq.SEZAINST where hlq is the high level qualifier data set name for TCPIP data sets.

3. Configure AT-TLS policy rules. The policy agent provides AT-TLS policy rules to the TCP/IP stack. Each rule defines a set of security conditions that the policy agent compares to the conditions at the connection that it is checking. When the policy agent finds a match, it assigns the connection to the actions that are associated with the rule.

4. Create a client keyring for each client with necessary certification authority certificates. The name of the client keyring should match the name that is provided for the keyring in the policy configuration file.

   If the web service user-defined functions are defined with Db2 security, the authorization ID that is associated with the WLM-established address space where the user-defined functions are defined must have all the necessary permissions to access the keyring.

5. Stop and start TCP/IP stacks.

6. Start the policy agent. Verify that message EZZ4248E is released from the console.

Results

Setup is complete and you can run the web service user-defined functions with HTTPS.

Enabling Db2 as a web service provider

Enabling Db2 as a web service provider allows you to create web services on z/OS with your Db2 data and applications.

About this task

Db2 for z/OS as a web service provider has the following prerequisites:

- Enable JDBC (legacy or universal) in Db2.
- Install WebSphere Application Server Version 5 or later on z/OS.
- Ensure that the WebSphere Application Server library contains the following two files:
  - mail.jar
  - activation.jar

  If mail.jar is not present, download JavaMail Version 1.2 or later. If activation.jar is not present, download JavaBeans Activation Framework Version 1.0.1 or later.

To use Db2 Web Services Object Runtime Framework (WORF), you need to make the run time services available to WebSphere Application Server (WAS). By default, WORF is installed in the HFS directory:

/usr/lpp/db2c10_worf/
The base installation directory contains the lib/ subdirectory that contains the runtime JAR file worf.jar. To begin using WORF, copy worf.jar, mail.jar, and activation.jar to a WAS shared library directory that you have already set up and restart WAS.

WORF provides a sample web application in the following directory:
lib/services.war

The application contains sample Document Access Definition Extension (DADX) files that define sample Db2 web services.

**Procedure**

To set up this application with sample DADX files:

1. Follow the instructions in the job prolog to customize and run job DSNTEJWS, which is located in the prefix.SDSNSAMP directory. DSNTEJWS creates the Db2 tables that are used by the sample application.

2. The sample application is set up to use universal JDBC driver connectivity to connect to a z/OS Db2 server. Configure WAS with JDBC providers that make use of universal JDBC driver and perform the following steps:
   a. Copy services.war to a temporary directory.
   b. Extract services.war by issuing the following command:
      ```
jar -xvf services.war
      ```
   c. Open the group.properties files, which are located in the following directories:
      ```
WEB-INF/classes/groups/dxx_sample
WEB-INF/classes/groups/dxx_travel
      ```
      Modify the dbDriver, dbURL userID, and password fields to have the following values:
      ```
dbDriver=com.ibm.db2.jcc.DB2Driver
dbURL=jdbc:db2://server:port/database
userID=DB2 userid
password=DB2 userid password
      ```
   d. Compress the files by issuing the following command:
      ```
jar -cvf services.war *
      ```
3. Use a web browser to connect to your WAS Administrative Console.
4. Under "Applications", select "Install New Application".
5. Select the "Server Path" option, and type the location of the services.war file in the text box. If you installed WORF in the default location, the server path is:
   ```
/usr/lpp/db2c10_worf/lib/services.war
   ```
6. Enter a context root for the application (for example, services). Click "Next".
7. On the following screens, respond as necessary for your local setup. You can accept the default settings.
8. On the last screen, click "Finish". Click "Save to Master Configuration" to apply your changes.

**What to do next**

To load the application:
1. On the WAS Administrative Console's main page, under "Applications", select "Enterprise Applications". Select the application and click "Start" to load the application.

2. After the application loads, point a browser to your server with the context root that you chose (for example, http://server:port/services/). The welcome page lists the sample DADX files that are provided in services.war. To test the services, click on the links.

**Enabling Db2 as a web service consumer**

Enabling Db2 as a web service consumer allows you to receive web service data in your Db2 applications.

**Before you begin**

Db2 for z/OS as a web service consumer has the following prerequisites:

- Install IBM XML Toolkit for z/OS 1.10
- Configure TCP/IP

**About this task**

Installation job DSNTIJMV contains sample startup procedure DSNWLM, which you can use as a model for your startup procedures. You must modify the startup procedure only if the web services routines are not defined to the DSNWLM_WEBSERVICES core WLM environment.

**Procedure**

To enable Db2 as a web service consumer:

1. Change the following items in each startup procedure:
   
   a. Change the procedure name from DSNWLM to the procedure name that you specified when you set up the WLM application environment.

   b. Change the value of APPLENV to the name of the WLM application environment that you set up for the web services consumer user-defined functions. The name of this environment must match the WLMENV parameter value in the configuration control statement for each stored procedure in job DSNTIJRT.

   c. Change the value of DB2SSN to your Db2 subsystem name.

   d. Add the data set name of the XML Toolkit load library (XPLINKed version) to the STEPLIB concatenation. If you used the default data set names when you installed the XML Toolkit, the load library data set name is userid.SIXMLOD1.

   e. Ensure that the Db2 SDSNLOAD2 library is added to the STEPLIB.

2. After you set up the WLM application environment, create a JCL startup procedure for the stored procedure address space.

**Creating REST user-defined functions**

A set of sample user-defined functions can be installed with Db2 to allow applications to access REST-based services through SQL. The basic REST scalar functions provide access to web-hosted content. Additional helper functions are available to provide URL encoding and decoding and Base64 encoding and decoding.
Before you begin

Ensure that you have the following prerequisites before you create the sample REST user-defined functions:

- Java JDK 5.0 or later
  The REST user-defined functions are implemented in Java and require this minimum JDK level.
- TCP/IP
  The REST user-defined functions require a functioning TCP/IP setup and access to the Internet.
- A Java WLM environment
  You can use the WLM environment that is supplied with Db2 for Java routines, DSNWLM_JAVA, or any existing Java WLM environment. By default, the jar file for the Db2 REST functions, db2restudf.jar, is installed in the same class path as other Java routines that are supplied with Db2: /usr/lpp/db2b10/base. If you installed db2restudf.jar in a different path, verify that this path is specified in the Language Environment variable file that is used by the WLM environment.

About this task

This task describes how to create the basic REST user-defined functions that are provided with Db2. It also describes how to optionally set up the REST user-defined functions to use SSL and how to enable HTTP proxy support for the REST user-defined functions.

Tip: The sample HTTP user-defined functions are intended to be used within Db2 SQL applications to access remote non-Db2 REST-based services through SQL statements. Do not confuse them with Db2 native REST services, which supports using a REST-based interface to interact with Db2 data from web, mobile, and cloud applications.

Procedure

To create and configure connections for the sample REST user-defined functions:

1. Modify job DSNTIJRF or the createUDFzOS.sql script in the db2restudf.jar file to specify the name of the WLM environment that you set up for the REST user-defined functions.
2. Run job DSNTIJRF or the createUDFzOS.sql script.
3. Optional: Enable SSL connections to use a key database or to use a RACF keyring.
   - To enable SSL connections to use a key database, specify the following settings in the file that you specified for JVMPROPS in the Language Environment options data set:
     - -Djavax.net.ssl.keyStore=/wlmjvm/keystore.jks
     - -Djavax.net.ssl.trustStore=/wlmjvm/keystore.jks
     - -Djavax.net.ssl.trustStorePassword=password
     - -Djavax.net.ssl.keyStorePassword=password
   - To enable SSL connections to use a RACF key ring, follow these steps:
     a. Create a key ring for the RACF user ID that is associated with the stored procedure address space in which the REST user-defined functions run.
        To create a key ring named RESTKEYRING for user ID RESTUSR, issue this command:
b. Add certificates to the key ring.

To add a certificate to RESTKEYRING, issue this command:

```
RACDCERT ID(RESTUSR) CONNECT(ID(RESTUSR) LABEL('RESTCERT') RING(RESTKEYRING))
```

c. In the file that you specified for JVMPROPS in the Language Environment options data set, specify the following settings:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.net.ssl.trustStore</td>
<td>Path to the key ring, in the format safkeyring://racf-id/ring-id</td>
</tr>
<tr>
<td>racf-id is the RACF user ID that is associated with the stored procedure address space. ring-id is the name of the RACF key ring.</td>
<td></td>
</tr>
<tr>
<td>javax.protocol.handler.pkg</td>
<td>com.ibm.crypto.provider</td>
</tr>
<tr>
<td>javax.net.ssl.trustStoreType</td>
<td>JCERACFKS</td>
</tr>
</tbody>
</table>

To set up SSL connections that use the RESTKEYRING key ring, set these properties:

- `-Djavax.net.ssl.trustStore=safkeyring://RESTUSR/RESTKEYRING`
- `-Djavax.protocol.handler.pkgs=com.ibm.crypto.provider`
- `-Djavax.net.ssl.trustStoreType=JCERACFKS`

4. Optional: Enable HTTP proxy support.

Set the following properties in the file that you specified for JVMPROPS in the Language Environment options data set:

- `http.proxyHost`
- `http.proxyPort`
- `http.proxyUser`
- `http.proxyPassword`
- `http.nonProxyHosts`

In this properties file, specify `-Dproperty=value` for each configuration property. For example, the following settings can be used:

- `-Dhttp.proxyHost=myProxyHost.com`
- `-Dhttp.proxyPort=8090`
- `-Dhttp.proxyUser=user`
- `-Dhttp.proxyPassword=secret`
- `-Dhttp.nonProxyHosts=*.*.ibm.com | wikipedia.org | ...`

Related concepts:

- [IBM Data Server Driver for JDBC and SQLJ support for SSL (Db2 Application Programming for Java)](https://www.ibm.com/support/knowledgecenter/SSEPGT_10.5.0/com.ibm.db2.udb.doc/doc/c0022071.html)

Related reference:

Sample SQL procedures for tracing native SQL routines

A set of sample SQL procedures that trace native SQL routines can be installed with Db2.

The trace SQL procedures and supporting objects are:

**DSN8.CREATE_DGTT**
An SQL procedure that creates declared temporary table SESSION.DBMS_BUFFER, which contains the trace message buffer.

**DSN8.DISABLE**
An SQL procedure that disables the SQL routine trace message buffer.

**DSN8.ENABLE**
An SQL procedure that enables the SQL routine trace message buffer.

**DSN8.GET_LINE**
An SQL procedure that returns a single line from the SQL routine trace message buffer. The returned data does not include an end-of-line character sequence.

**DSN8.GET_LINES**
An SQL procedure that returns one or more lines from the SQL routine trace message buffer, and stores the lines in an array. The returned data does not include end-of-line character sequences.

**DSN8.NEW_LINE**
An SQL procedure that writes an end-of-line character sequence to the SQL routine trace message buffer.

**DSN8.PUT**
An SQL procedure that writes a line to the SQL routine trace message buffer, without an end-of-line character sequence.

**DSN8.PUT_LINE**
An SQL procedure that writes a line to the SQL routine trace message buffer, with an end-of-line character sequence.

**DSN8.DBMS_ENABLE**
A global variable that is used to test whether the SQL routine trace message buffer is enabled.

**DSN8.GRPHICARR**
A user-defined array type that can be used to define the array into which the output from DSN8.GET_LINES is written.

Run the provided installation job DSNTEJTR to create the SQL procedures for tracing native SQL routines. See the DSNTEJTR job prolog for instructions on how to customize the job for your site.

Related reference:

- [DSN8.DISABLE stored procedure (Db2 SQL)]
- [DSN8.ENABLE stored procedure (Db2 SQL)]
- [DSN8.GET_LINE stored procedure (Db2 SQL)]
- [DSN8.GET_LINES stored procedure (Db2 SQL)]
- [DSN8.NEW_LINE stored procedure (Db2 SQL)]
- [DSN8.PUT stored procedure (Db2 SQL)]
Sample IBM InfoSphere BigInsights user-defined functions

A set of sample user-defined functions can be installed with Db2 to allow applications to access IBM InfoSphere® BigInsights® data through SQL.

The IBM InfoSphere BigInsights user-defined functions read data in the Hadoop Distributed File System (HDFS), and submit Jaql queries.

Related reference:
- [HDFS_READ (Db2 SQL)]
- [JAQL_SUBMIT (Db2 SQL)]

Creating sample IBM BigInsights user-defined functions

You can customize and run job DSNTEJBI to create the sample IBM BigInsights user-defined functions.

Before you begin

Ensure that you have the following prerequisites before you create the sample IBM BigInsights user-defined functions:
- IBM BigInsights 3.0, with HDFS
- A supported z/OS C++ compiler
- A WLM environment for web services

You can use the WLM environment that is supplied for web service user-defined functions, DSNWLM_WEBSERVICES, or any WLM environment with similar attributes. DSNWLM_WEBSERVICES is the default WLM environment for the IBM BigInsights user-defined functions. If you use another WLM environment, you need to update the WLM environment name in the CREATE FUNCTION statements for those user-defined functions.

Procedure

To create the sample IBM BigInsights user-defined functions:
1. Modify job DSNTEJBI to as indicated in the job prolog.
2. Run job DSNTEJBI.

Related concepts:
- Job DSNTEJBI

Job DSNTEJBI

DSNTEJBI defines and prepares the IBM InfoSphere BigInsights sample user-defined functions.

The installation CLIST does not customize job DSNTEJBI. Follow the instructions in the DSNTEJBI prolog to manually customize the job.

The IBM InfoSphere BigInsights sample user-defined functions are:
Function | Description
--- | ---
HDFS_READ | Reads data from a delimiter-separated file in the Hadoop Distributed File System (HDFS).
JAQL_SUBMIT | Invokes an IBM InfoSphere BigInsights Jaql query.

Job DSNTEJBI contains the steps that are listed in the table below.

Table 133. Steps in job DSNTEJBI

<table>
<thead>
<tr>
<th>Step</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drops the SYSFUN.HDFS_READ and SYSFUN.JAQL_SUBMIT user-defined functions.</td>
</tr>
<tr>
<td>2</td>
<td>Drops the SYSFUN.HDFS_READ and SYSFUN.JAQL_SUBMIT user-defined functions.</td>
</tr>
<tr>
<td>3</td>
<td>Grants access to the SYSFUN.HDFS_READ and SYSFUN.JAQL_SUBMIT user-defined functions.</td>
</tr>
<tr>
<td>4</td>
<td>Prepares program DSN8GLIB, which contains functions for accessing generic table functions.</td>
</tr>
<tr>
<td>5</td>
<td>Prepares program DSN8HLIB, which contains functions for HTTP access.</td>
</tr>
<tr>
<td>6</td>
<td>Prepares program DSN8HDFS, the external module for user-defined function SYSFUN.HDFS_READ.</td>
</tr>
<tr>
<td>7</td>
<td>Prepares program DSN8JAQL, the external module for user-defined function SYSFUN.JAQL_SUBMIT.</td>
</tr>
</tbody>
</table>

**Note:**
1. This is an optional step. You need to uncomment this step before you can run it.
2. This is an optional step that is required only if you run either of the steps that follow it. You need to uncomment this step before you can run it.

If DSNTEJBI runs successfully, it produces the return codes that are shown in the table below.

Table 134. DSNTEJBI return codes

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEJBIS01</td>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>TEJBIS02</td>
<td>0000 or 0004</td>
<td></td>
</tr>
<tr>
<td>TEJBIS03</td>
<td>0000 or 0004</td>
<td></td>
</tr>
<tr>
<td>TEJBIS04</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKELD</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>TEJBIS05</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKELD</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
<tr>
<td>TEJBIS06</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKELD</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>
Table 134. DSNTEJBI return codes (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>PROCSTEP</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEJBIS07</td>
<td>PC</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>CP</td>
<td>0000</td>
</tr>
<tr>
<td></td>
<td>PLKED</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>0000</td>
</tr>
</tbody>
</table>

Notes:
1. The PLKED step does not exist in the job if the value of the CPP LE PRELINKER PHASE field is set to NO.

Related concepts:
- Sample user-defined functions (Db2 SQL)
- Steps to creating and using a user-defined function (Db2 Application programming and SQL)

The IBM Data Server Driver for JDBC and SQLJ

If you plan to run JDBC or SQLJ applications, you need to install the current version of the IBM Data Server Driver for JDBC and SQLJ. You must complete this task after you install Db2 for z/OS or migrate to the current version of Db2 for z/OS.

Related tasks:
- Installing the IBM Data Server Driver for JDBC and SQLJ on Db2 for z/OS (Db2 Application Programming for Java)

Connecting the IMS attachment facility

Connecting Db2 to IMS requires coordination with your company’s group that supports IMS systems.

Procedure

To connect the IMS attachment facility:
- Make Db2 load modules available to IMS
- Define Db2 to IMS
- Define new programs and transactions to IMS
- Depending on your site, you might also need to:
  - Define Db2 plans for IMS applications
  - Generate a user language interface

What to do next

An IMS system definition might be required to complete the steps to connect Db2 to IMS. If RACF is installed, you also need to define the IMS-to-Db2 connection to RACF.

Related tasks:
- Defining Db2 resources to RACF (Managing Security)
Making Db2 load modules available to IMS

As part of connecting Db2 to IMS, you must make the Db2 load modules available to IMS.

About this task

If you have already included the prefix.SDSNLOAD library in your LNKLSTxx, you can skip this step. Db2 11 modules will be available through normal z/OS module search.

Procedure

To make Db2 load modules available to IMS:

- If Db2 load libraries are not included in your LNKLSTxx, add STEPLIB statements to your startup procedures.
- Add prefix.SDSNLOAD to the DFSESL DD statement.
- If any IMS region connects to more than one release of Db2, then you must ensure that the Db2 load library that is used for that region is compatible with each release. The IMS attachment facility is upward compatible, but not downward compatible. This means that you should use the oldest release of the Db2 load library for the IMS region.
  - If all the data sets referred to in the JOBLIB or STEPLIB statement for an IMS region are APF-authorized, then add the DD statement for prefix.SDSNLOAD to the JOBLIB or STEPLIB statement. If the DYNAM option of COBOL is being used, the IMS RESLIB DD statement must precede the reference to prefix.SDSNLOAD in the JOBLIB or STEPLIB statement.
  - Add the ddname DFSESL DD statement for prefix.SDSNLOAD. All libraries specified on the DFSESL DD statement must be APF-authorized. The DFSESL DD statement is not required by Db2 DL/I batch support. IMS requires that an IMS RESLIB DD statement also be referenced by the DFSESL DD statement, as in the following:

```plaintext
//DFSESL DD DSN=ims_reslib,DISP=SHR
//DD DSN=prefix.SDSNLOAD,DISP=SHR
```

Defining Db2 to IMS

To define Db2 to IMS, the Db2 identification must be defined to the control region and the DL/I batch region. Optionally, you can also define the Db2 identification to each dependent region that accesses the Db2 subsystem.

Procedure

To define Db2 to IMS:

1. Name an SSM member in IMS.PROCLIB by concatenating the value of the IMSID field of the IMS IMSCTRL macro (one to four alphanumeric characters) with any name (one to four alphanumeric characters) defined by your site.

   Each SSM entry in IMS.PROCLIB defines at least one connection from an IMS region to at least one different z/OS subsystem. One SSM member can be shared by all of the IMS regions, or a specific member can be defined for each region. This record contains as many entries as there are connections to external subsystems. Each entry is an 80-character blocked or deblocked record. Fields are keyword or positional and are delimited by commas. The fields in this record are:
SST=,SSN=,LIT=,ESMT=,RTT=,REO=,CRC=

where:

**SST=**
Identifies a required one-to eight-character name that defines the external subsystem type. It must be set to DB2 for IMS to connect to Db2.

**SSN=**
Identifies a required one-to four-character Db2 subsystem name. This name must be the name you specified for SUBSYSTEM NAME on installation panel DSNTIPM. The default is DSN1.

**LIT=**
Identifies a required four-character alphanumeric option, specifying the language interface token (LIT) supplied to IMS. The IMS-supplied language interface module (DFSLI000) requires a value of SYS1 for this option.

**ESMT=**
Identifies a required one-to eight-character alphanumeric option specifying the external subsystem module table. This module specifies which attachment modules must be loaded by IMS. **DSNMIN10** is the required value for this field.

**RTT=**
Identifies an optional one to eight character alphanumeric name of the user-generated resource translation table (RTT). This table maps the IMS application names into Db2 plan names. If this entry is omitted, the Db2 plan name is the IMS application load module name.

**REO=**
Identifies the optional one-character region error option to be used if an IMS application attempts to reference a non-operational external subsystem or if resources are unavailable at create thread time. If Db2 detects the unavailable resource condition during normal SQL processing, a -904 SQLCODE is returned to the application.

**R**
Passes a SQL return code to the application, indicating that the request for Db2 services failed (default). The most commonly returned SQL codes are -922, -923, and -924. However, there might be other SQL codes returned to the application.

When the first connection to Db2 cannot be established, a SQL return code is not returned. Instead, the application is terminated with an abend code U3047.

**Q**
Abends the application with an abend code U3051, backs out activity to the last commit point, does a PSTOP of the transaction, and re-queues the input message. This option only applies when an IMS application attempts to reference a non-operational external subsystem or if the resources are unavailable at create thread time. If Db2 detects the unavailable resource condition during normal SQL processing, a -904 SQLCODE is returned to the application.

**A**
Abends the application with an abend code of U3047 and discards the input message. This option only applies when an IMS application attempts to reference a non-operational external subsystem or if the resources are unavailable at create thread time. If Db2 detects the unavailable resource condition during normal SQL processing, a -904 SQLCODE is returned to the application.
If Db2 is not active or the connection cannot be established when the first SQL call is made from the application program (such as Db2 unavailable, Db2 quiescing, or Db2 terminating), the action you take depends on the region error option specified. SQL codes of -922, -923, or -924 might be returned to the application if option R is specified.

You can change the default for an application if a resource translation table entry is generated for that application.

**CRC**
Identifies a command recognition character used by IMS to identify Db2 commands entered from an IMS terminal with the /SSR command. Any character is valid for the CRC except the period (.), slash (/), or comma (,). The default CRC is the hyphen (-).

These options apply to DL/I batch only:

**CONNECTION_NAME**
Represents the name of the job step that is the coordinator for Db2 activity. The connection name is optional. The connection name defaults are:

<table>
<thead>
<tr>
<th>Type of Application</th>
<th>Default Connection Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch job</td>
<td>Job name</td>
</tr>
<tr>
<td>Started task</td>
<td>Started task name</td>
</tr>
<tr>
<td>TSO user</td>
<td>TSO authorization ID</td>
</tr>
</tbody>
</table>

If a batch job fails, you must use a separate job to restart the batch job. The connection name used in the restart job must be the same as the name used in the batch job that failed. Or, if the default connection name is used, the restart job must have the same job name as the batch update job that failed.

Db2 requires unique connection names for Db2 DL/I batch support. If two applications try to connect with the same connection name, then the second application is not allowed to connect to Db2. CONNECTION_NAME can be 1 to 8 characters long.

**PLAN**
Identifies a Db2 plan name. If you do not specify a plan name, the application program module name is checked against the optional resource translation table. If a match is found, the translated name is used as the Db2 plan name. If no match is found, the application program module name is used as the plan name. PLAN can be 1-8 characters long.

**Important:** If you use the External Subsystem Attach Facility (ESAF) and the IBM Data Server Driver for JDBC and SQLJ to connect to a Db2 subsystem, a set of JDBC driver packages with a collection name that matches the application program module name (PSB name) must exist. If no packages with a matching collection name exist, you need to bind a set of driver packages with the -collection parameter value set to the application program module name. See [DB2Binder utility (Db2 Application Programming for Java)](https://www.ibm.com/support/knowledgecenter/SSEPGH_10.1.0/com.ibm.db2.luw.udf.sqlreference.doc/doc/com.ibm.db2.luw.udf.sqlreference.doc.doc.html) for information on binding JDBC driver packages.

**PROG**
Identifies the name of the application program that is to be loaded and to receive control. PROG is required and can be 1-8 characters long.

2. Provide support for entering Db2 commands through IMS by defining the command recognition character (CRC) in the SSM member of the IMS control region. If the /SSR command is entered through the z/OS console, the
AUTHID WT OR needs to be granted the appropriate authority. If the /SSR command is entered through an IMS terminal, the IMS LTERM name or the signon ID (if active) needs to be granted the appropriate authority. The /SSR command format is:

/SSR crc DB2 command

as in

/SSR -DISPLAY THREAD (*)

3. Specify the SSM EXEC parameter in the startup procedure of the IMS control, MPP, BMP, or DL/I batch region. The SSM is concatenated with the IMSID to form a member name in IMS.PROCLIB. The IMSID comes from the IMSID option of the IMSCTRL generation macro or the IMSID option in the control region startup procedure.

For DL/I batch regions, you can specify the Db2 connection parameters in the DDITV02 data set instead of an SSM member. The DDITV02 data set and an SSM member have the same format.

If you specify the SSM for the IMS control region, any dependent region running under the control region can attach to the Db2 subsystem named in the IMS.PROCLIB member specified by the SSM parameter. The IMS.PROCLIB member name is the IMS ID (IMSID=xxxx) concatenated with the one to four characters specified in the SSM EXEC parameter. The IMS ID is the IMSID parameter of the IMSCTRL generation macro.

IMS allows you to define as many external subsystem connections as are required. More than one connection can be defined for different Db2 subsystems. All Db2 connections must be within the same z/OS system. For a dependent region, you can specify a dependent region SSM or use the one specified for the control region. You can specify different region error options (REOs) in the dependent region SSM member and the control region SSM member. The following table shows the different possibilities of SSM specifications.

<table>
<thead>
<tr>
<th>SSM for Control Region</th>
<th>SSM for Dependent Region</th>
<th>Action</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>None</td>
<td>No external subsystem can be connected</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>None</td>
<td>No external subsystem can be connected.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Use the control region SSM</td>
<td>Applications scheduled in the region can access external subsystems identified in the control region SSM. Exits and control blocks for each attachment are loaded into the control region address space.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes (NULL entry)</td>
<td>No SSM is used for the dependent region</td>
<td>Applications scheduled in this region can access DL/I databases only. Exits and control blocks for each attachment are loaded into the control region address space and each dependent region address space.</td>
</tr>
</tbody>
</table>
### Table 136. SSM specifications options (continued)

<table>
<thead>
<tr>
<th>SSM for Control Region</th>
<th>SSM for Dependent Region</th>
<th>Action</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Check the dependent region SSM with the control region SSM.</td>
<td>Applications scheduled in this region can access only external subsystems identified in both SSMs. Exits and control blocks for each attachment are loaded into the control region and the dependent region address space.</td>
</tr>
</tbody>
</table>

No specific parameter exists to control the maximum number of SSM specification possibilities.

**Related concepts:**
- [Input and output data sets for DL/I batch jobs (Db2 Application programming and SQL)](#)
- [Items to include in a batch DL/I program (Db2 Application programming and SQL)](#)
- [System definition](#)

**Related tasks:**
- [Defining Db2 plans for IMS applications (optional)](#)

### Defining new programs and transactions to IMS

You can define new programs and transactions that access Db2 resources to your IMS system.

**About this task**

Coordinate with your IMS support group to install the programs and transactions for Phase 4 of the verification process.

Programs and transactions already defined to IMS can use SQL without any additional definition to IMS.

**Related tasks:**
- [Verifying successful Db2 12 installation or migration with sample applications](#)

### Defining Db2 plans for IMS applications (optional)

The application plan defines the Db2 resources that an application accesses.

**About this task**

The application plan is identified by its plan name. Each IMS application is associated with a plan name.

The default is to have the Db2 plan name the same as the IMS application program load module name. If you assigned a different name to the plan, you need a resource translation table (RTT). If you chose an error option different from the REO default, you also need an RTT. Db2 provides the DSNMAPN macro in `prefix.SDSNMACS` to generate an RTT. After it is assembled, the table must be link-edited as REENTRANT with RMODE=24 into any authorized library that is concatenated with the library from which IMS loads the Db2 IMS attach modules.
The format of DSNMAPN macro is shown in the following table.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNMAPN</td>
<td>APN=</td>
<td>IMS application name</td>
</tr>
<tr>
<td></td>
<td>PLAN=</td>
<td>Associated Db2 plan name</td>
</tr>
<tr>
<td></td>
<td>[,OPTION=]</td>
<td>Specific entry error option R, Q, or A. See REO in the SSM entry.</td>
</tr>
<tr>
<td></td>
<td>[,END=]</td>
<td>Indicates last entry (YES/NO). NO is the default.</td>
</tr>
</tbody>
</table>

Related concepts:
- IMS attachment facility macro (DSNMAPN)
- Items to include in a batch DL/I program (Db2 Application programming and SQL)

**IMS language interface module**

You must generate a user language interface only if you intend to access two Db2 subsystems from the same dependent region.

To provide this access, the subsystem member (SSM) must contain one entry for each subsystem. Each entry contains a different subsystem ID and its associated language interface token (LIT). IMS provides the DFSLI macro to generate additional language interface modules with unique LITs. The general format of the macro is shown in the table below.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFSLI</td>
<td>TYPE</td>
<td>Specifies the type of subsystem that can be accessed through this language interface module. Db2 is the only value supported by this option</td>
</tr>
<tr>
<td></td>
<td>LIT</td>
<td>Defines a name (called LIT) to relate a language interface module with an entry in the SSM for the dependent region</td>
</tr>
</tbody>
</table>

When an IMS application issues a Db2 request, IMS knows the target subsystem by the LIT used in the request. For example, consider the case of a dependent region accessing two Db2 subsystems (DSN1 and DSN2):

- You generate a language interface with LIT=SYS2 (DFSLI001).
- You define two entries in the SSM member. The first entry points to DSN1 with LIT=SYS1; the second points to DSN2 with LIT=SYS2.
- You link-edit applications accessing the DSN1 subsystem with the IMS-provided language interface (DFSLI000).
- You link-edit applications accessing the DSN2 subsystem with the user-generated language interface (DFSLI001).

Even though a region can communicate with two or more Db2 subsystems, an IMS application can access only one—the Db2 subsystem referred to in the language interface that is link-edited. You can alter the SSM to route application requests to a different Db2 subsystem.
IMS attachment facility macro (DSNMAPN)

The DSNMAPN macro is required when the IMS application load module name is different than the name of its related IBM Db2 application plan. The macro is also required if the error option is different than the ERR value that is specified on the IMS SSM entry.

Macro statements are assembled in prefix. SDSNMACS and must be link-edited as REENTRANT with RMODE=24 into the Db2 library prefix. SDSNLOAD. The module name must be specified on the IMS SSM entry for the Db2 subsystem. The name must be specified as in the RTT entry for the SSM member defining the connection of this region. IMS loads the RTT module into the dependent region address space.

Note:
1. The macro name must be followed by one or more blanks before options are coded.
2. Multiple options must be separated by commas (with no blanks).

**label DSNMAPN**

DSNMAPN is the name of the macro. It must be coded exactly as it appears here, and it must be separated from any optional options by one or more blanks.

For **label**, substitute the CSECT name of your module. This name must match the name of the module specified to the linkage editor. **Label** is optional except for the first invocation of the DSNMAPN macro. The last invocation requires END=YES.

**APN=program-name**

Specifies the name of an application load module scheduled by IMS. For **program-name**, substitute an application name of up to eight characters.

**PLAN=plan-name**

Specifies an application plan name that is used (instead of the default application name) when a thread is created. For **plan-name**, substitute an application plan name of up to eight characters.

**OPTION=R|Q|A**

Specifies the action taken when an application program call cannot be performed because there is some problem in communication between the application program and the Db2 subsystem or if resources are unavailable. The value can be R, Q, or A.

If **OPTION** is not specified, the region error option (REO) is used.

**R** Specifies that a return code is returned to the application to indicate that the request for Db2 services failed.

**Q** Specifies that the transaction is abnormally terminated with an abend code U3051, activity is backed out to the last commit point, and the input message is re-queued.

**A** Specifies that the transaction is abended with an abend code of U3047, and the input message is deleted.

Default: R

**END=NO|YES**

Specifies whether this is the last DSNMAPN macro invocation. The value can be NO or YES.
NO Specifies that this is not the last DSNMAPN macro invocation.

YES Specifies that this is the last DSNMAPN macro invocation.

Default: NO

The last DSNMAPN macro invocation must be followed by the specification END=YES.

Usage notes
• To enter more than one application name (with its corresponding plan name and OPTION specification), you must use multiple invocations of the DSNMAPN macro. The first invocation requires the label; the last invocation requires END=YES.
• Invocations must be in ascending order by application name. If they are not, an MNOTE macro error is generated.

Related tasks:
[Defining Db2 plans for IMS applications (optional)]

Connecting distributed database systems

You can use the distributed data facility (DDF) of Db2 to access data that is held by other data management systems. You can also use DDF to make your Db2 data accessible to other systems.

About this task

Db2 does not place any upper limit on the number of systems it can connect to; available storage is the limiting factor.

Related concepts:
[Communicating with data sharing groups (Db2 Data Sharing Planning and Administration)]

The communications protocols

DDF uses TCP/IP or SNA to communicate with other systems.

The figure below shows the connectivity options that you have with DDF in Db2.
Setting up a network for use by database management systems requires knowledge of both database management and communications. Thus, you must put together a team of people with those skills to plan and implement the network.

**The role of the communications database (CDB)**

When sending a request, Db2 uses the LINKNAME column of the SYSIBM.LOCATIONS catalog table of the communications database (CDB) to determine which protocol to use.

To receive VTAM requests, you must specify an LUNAME on installation panel DSNTPR. To receive TCP/IP requests, you must select a DRDA port and a resynchronization port on installation panel DSNTP5. TCP/IP uses the server’s port number to pass network requests to the correct Db2 subsystem.
If the value in the LINKNAME column is found in the SYSIBM.IPNAMES table, TCP/IP is used for DRDA connections. If the value is found in SYSIBM.LUNAMES table, SNA is used. If the same name is in both SYSIBM.LUNAMES and SYSIBM.IPNAMES, TCP/IP is used.

Attention: A requester cannot connect to a given location using both SNA and TCP/IP protocols. For example, if your SYSIBM.LOCATIONS table specifies a LINKNAME of LU1, and LU1 is defined in both the SYSIBM.IPNAMES and SYSIBM.LUNAMES table, TCP/IP is the only protocol used to connect to LU1 from this requester for DRDA connections.

The DRDA database protocol

Applications can use the Distributed Relational Database Architecture™ (DRDA) protocol to control remote access.

With DRDA, the application connects to a server at another location and executes packages that have been previously bound at that server. The application uses a CONNECT statement, a three-part name, or an alias to access the server.

Queries can originate from any system or application that issues SQL statements as a requester in the formats required by DRDA.

For two-phase commit using Systems Network Architecture (SNA) connections, Db2 supports both presumed abort and presumed nothing protocols that are defined by DRDA. If you are using TCP/IP, Db2 uses the sync point manager that is defined in the documentation for DRDA Level 3.

Related concepts:

- Types of Two-Phase Commit (z/OS MVS Programming: Resource Recovery)

Related information:

- The Open Group

**DDF installation and implications for Db2**

The installation options for DDF are described on installation panels DSNTIPR and DSNTIP5.

Use these options to define, among other things:

- Whether you want DDF to start automatically when Db2 starts
- Important names for this Db2 subsystem, including a LU name and NETID
If you plan to use both VTAM and TCP/IP, you must define an LU name and NETID. Db2 as a requester generates the unit of work using NETID and LUNAME.

- Thread management options
- Security options
- TCP/IP port numbers
- Control of the number of DRDA query blocks that can flow on a network request that was specified with OPTIMIZED FOR n ROWS where \( n \) exceeds the number of rows that fit in a single query block.

If you do not plan on using SNA/APPC communications, you can set up DDF to not activate its SNA/APPC support. To set up DDF to not activate its SNA/APPC support, give the Db2 subsystem an IPNAME value by updating the BSDS DDF record. Setting up DDF to not activate its SNA/APPC support is not supported by the Db2 installation option panels. Db2 TCP/IP communications uses the IPNAME value and a character representation of the TCP/IP resync port (RESPORT) hexadecimal value to identify units of work.

**Support for extended dynamic SQL:** If this Db2 subsystem services requesters that support extended dynamic SQL, such as DB2 Server for VSE & VM, enter YES in field DESCRIBE FOR STATIC on installation panel DSNTIPF. This option lets applications from the requesting system execute SQL DESCRIBE statements that appear as extended dynamic SQL statements in the requesting system, but appear as static SQL in the Db2 package. For the option to take effect, you must bind the package with DESCRIBE FOR STATIC enabled.

**Test your connections** You should test systems with each other to ensure that their communications setups are correct. If you are testing with another Db2 for z/OS, enter the location name of that other site in field REMOTE LOCATION of installation panel DSNTIPY. The remote location must also have DDF installed and active and must have run the first sample job, DSNTEJ1.

**Related reference:**
- DSNTIPR: Distributed data facility panel 1
- DSNTIP5: Distributed data facility panel 2
- DSNJU003 (change log inventory) (Db2 Utilities)

### Connecting systems with VTAM

*Virtual Telecommunications Access Method* (VTAM) is a program that you can use to control remote Db2 communication.

#### About this task

Remote communications terminology:

**Logical unit (LU)**

A source of requests entering the network and a receptor of replies from the network. For example, a particular Db2 is an LU.

**Session**

A logical connection between two LUs. Multiple sessions can run on a single physical connection.
Conversation
A dialog that uses a session to transfer information between transaction programs, such as Db2 to Db2. A single session can support multiple conversations, but only one at a time.

To prepare Db2 for communication using VTAM and the distributed data facility (DDF), complete the following steps. You can complete steps 1, 2, and 3 after installing Db2. Steps 6 through 8 are optional.

• **Step 1: Customize VTAM for Db2**
To make monitoring of the network easier, consider installing NetView®.

• **Step 2: Choose names and a password**
You need to choose two names for the local Db2 subsystem: a location name and a logical unit name (LU name).

A location name distinguishes a specific database management system in a network, so applications use this name to direct requests to your local Db2 subsystem. Other systems use different terms for a location name. For example, Db2 Connect calls this the target database name. The DRDA term, RDBNAM, is used to refer to non_Db2 systems’ relational database names.

An LU name is the name by which VTAM recognizes this subsystem in the network. You might need to know the LU names of other systems that can request data from the local Db2 subsystem, or you can use a default LU name of eight blanks.

If you plan to request data from other systems, you need the LU names and location names for those serving systems. Most of the time, system administrators and operators need to know both names, because they can use both names in various commands, and Db2 uses both names in messages.

In addition to the names mentioned above, you can choose an optional password to validate your local Db2 subsystem to VTAM. If the z/OS system on which Db2 is running is part of an z/OS Parallel Sysplex, you can choose a generic LU name to define a Db2 group to remote locations.

• **Step 3: Define the Db2 subsystem to VTAM**
Use the VTAM APPL statement to make the Db2 subsystem known to VTAM. You must include the APPL definitions in the VTAM SYS1.VTAMLST library at VTAM startup.

Also use the VTAM MODEENT statement to define default session modes. You must include mode tables in the VTAM SYS1.VTAMLIB library at VTAM startup.

Sample VTAM definitions are provided in the data set DSN8VTAM in SDSNSAMP and in examples throughout this topic.

• **Step 4: Populate the communications database**
The Db2 catalog includes the communications database (CDB), which contains several tables that hold information about your connections with remote systems. You must populate some of these tables before you can request data from those remote systems. If this Db2 system only services data requests, you do not have to populate the CDB; you can use the default values.

• **Step 5: Start VTAM to use Db2**
When you start VTAM to use Db2, you must be sure that the proper definitions are in the VTAM libraries VTAMLST and VTAMLIB.

• **Step 6: Tune the system**
This is an optional step, which you can do after you have established communications between two or more systems. The procedure outlined up to
this point gives you default values for your Db2 modes and your class of service. Although the defaults are probably adequate for your preliminary testing, you can change them to improve performance in the network, or to assign different modes to different application plans.

When VTAM links two nodes, it establishes a session. The number of available sessions can have a significant impact on performance; therefore, you might need to modify your session limit values. Also, large amounts of Db2 data travelling through the network can severely affect VTAM storage, and you might need to tune buffer storage.

You can also tune the system by changing mode options. A mode describes various characteristics of a session, such as the maximum number of bytes sent at one time. Modes can point to a class of service table, which ranks the available virtual routes for this mode with respect to preference of use and paths through the network. Essentially, the class of service table allows you to assign different network priorities to your modes.

• "Step 7: Create Aliases"
  This is an optional step. Each Db2 location can create aliases for the tables it wants to access, using DRDA, at the other Db2 locations.

• "Step 8: Provide Authorization for an Appropriate Level of Security"

Related concepts:
Sample VTAM definitions to connect two Db2 subsystems

Related tasks:
Calculating session limits
Calculating VTAM I/O buffer pool (IOBUF) storage

Related reference:
IBM Tivoli NetView for z/OS (Installation: Getting Started)
Generic resources (SNA Network Implementation Guide)
Implementing a VTAM network (SNA Network Implementation Guide)
Tuning VTAM for your environment (SNA Network Implementation Guide)

Customize VTAM for Db2

For Db2 to provide the best performance for distributed database systems, you probably need to customize VTAM.

Before you customize VTAM, consider the communication needs of your Db2 connections. Because you could allow your Db2 subsystem to send large amounts of data through the network, reexamine the capacity of your existing network. In some cases, portions of your existing network might need additional communication hardware to provide the required capacity.

Related reference:
SNA Network Implementation Guide

Choosing names and a password

When you connect a Db2 subsystem to VTAM, you must choose names for your local Db2 subsystem and, possibly, a VTAM password for it. You might also need to know the names of remote systems in the network.
Choosing names for the local subsystem

You define the names for the local subsystem and its VTAM password to Db2 by using the installation panels, or by using the change log inventory utility.

Procedure

Choose the following names for the local Db2 subsystem:

- A unique name by which the other systems in the network can recognize your subsystem. The name can have from 1 to 16 characters and is called the location name. (Db2 Connect refers to this as the target database name.) Make sure that the local location name is different from the name of every other system in the network, no matter where it is physically located.

  You must share the location name with the other systems that need to send SQL requests to this one.

  The location name should not change even if the network changes. Therefore, tightly control the allocation of location names. To ensure uniqueness, you use an IBM-registered SNA NETID as the first six bytes of your location name. If location names are not unique, you have to change many programs and tables if your network is later joined with another network using the same location name.

  The IBM recommendation for the NETID is the following format:

  - The first two bytes are the country code as defined in ISO standard ISO 3166. These codes include the uppercase letters A through Z.

  - The next four bytes are the enterprise code of the registering enterprise. This might already be registered with IBM as your SNA NETID. The enterprise code can include the uppercase letters A through Z, the numbers 0 through 9, and the underscore character (_).

  To register your SNA NETID, see your IBM representative.

- A name by which VTAM can recognize the local subsystem. It must be either a unique name or, in some cases, a generic name.

  - The unique name must be unique within the network of connected systems, can have from 1 to 8 characters, and is called the LU name. The LU name and the location name of a subsystem can be identical, but this is not recommended; LU names are unique only within a network, and networks can change. You must share the LU name with any system that requests data from your local subsystem. Later, you enter this name in the VTAM APPL statement.

  - If the z/OS system on which Db2 is running is part of a z/OS sysplex, you can use a generic 8-character name to represent a group of VTAM LU names. The generic name might be useful if your network is in a transitional period, and you want to use generic names to reference network nodes.

    Specify the generic LU name in the field Db2 GENERIC LUNAME on installation panel DSNTIPR. Use column GENERIC of SYSIBM.LUNAMES to indicate that you want to use the generic LU name for CNOS processing and SQL requests to a particular server.

- Server location aliases. You can use the Change Log Inventory utility to define up to eight aliases for a location. Applications can use these alias names to refer to the local Db2 subsystem or data sharing group.

  Aliases are most useful in a data sharing environment in which two or more Db2 subsystems are migrated to a single data sharing group. In this case, you can define the old location for each subsystem as an alias for the location name of the group. Remote applications that refer to the old location names do not need to change.
Use the Print Log Map utility to print all location alias names that are defined for a Db2 subsystem.

You can set up multiple server location aliases as locations on the z/OS requester to restrict which members an application will use when accessing a data sharing group. The z/OS requester must be using the IPLIST or LULIST table to route connections to the data sharing group.

**Related concepts:**
- SYSIBM.LUNAMES (Db2 Data Sharing Planning and Administration)

**Related tasks:**
- Defining the Db2 subsystem to VTAM
- Using the change log inventory utility to update the BSDS

**Choosing a password for the local subsystem**

Choosing a VTAM password is optional. Db2 does not require you to use a password as long as you have not included one in the VTAM APPL statement.

**Procedure**

To set a local subsystem password:

Enter the password by using the PRTCT option of the VTAM APPL statement. The password can have from one to eight EBCDIC characters.

This password is not transmitted through the network, so there is no need to share the password with the other systems.

**Related tasks:**
- Controlling LU attachments to the network (Managing Security)

**Names that you need from the remote systems**

If you plan to have your Db2 subsystem request data from other systems, you need the LU names and location names for those serving systems.

**Location names and LU names:** When you populate the communications database (CDB) in the local Db2, you must know the location names (or DRDA RDBNAMs) and LU names of remote servers. The remote servers are the systems from which this Db2 system is to request data. The local Db2 system does not need location names of requesters. However, you need to know the LU names of the requesters if you intend to change default communication options.

Db2 does not receive DRDA RDBNAM from requesters other than Db2 for z/OS. If Db2 does not have an RDBNAM, it displays LU names in messages, display output, and trace output. To help you distinguish between location names and LU names in those cases, the LU name is enclosed in less-than (<) and greater-than (>) brackets.

When your systems begin communicating, you and others involved in working with distributed systems need to be aware of the LU name to DRDA RDBNAM mappings. When you have obtained the necessary names, enter them in the CDB.

**Transaction program names (TPNs):** If a server is not a Db2 for z/OS, it might have an additional name that uniquely identifies it. In LU 6.2, this is known as a transaction program name (TPN), and can be from 1 to 64 characters long. When a
Db2 for z/OS subsystem communicates with other Db2 for z/OS subsystems, you do not need to supply TPN values. The Db2 subsystems automatically choose the correct TPN values.

**When a TPN is necessary:** You might need to supply TPN values when a Db2 subsystem requests data from a server that is not a Db2 for z/OS subsystem. For cases where the server does not accept the default TPN for DRDA access, enter into your CDB the TPN chosen by that server. For Db2 for VM, for example, the TPN is the SQL database machine ID.

**TPN values accepted by Db2 for z/OS:** A requester that is not Db2 for z/OS must use either the TPN name X'07F6C4C2' or DB2DRDA, which are the only values Db2 recognizes when it accepts a request from another system. Some requesters enter the TPN as two separate fields: a 1-byte prefix (X'07') and a 3-byte suffix ('6DB').

**Related tasks:**
- Populating the communications database for use with TCP/IP

**Names that Spiffy Computer Company chose**
Spiffy has chosen location names and LU names, some of which are used in the provided examples.

**Table 139. Spiffy's location names, LU names, and transaction program names (TPNS)**

<table>
<thead>
<tr>
<th>Location Name</th>
<th>LU Name</th>
<th>TPN</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>USIBMSTODB21</td>
<td>LUDB21</td>
<td></td>
<td>Db2 *</td>
</tr>
<tr>
<td>USIBMSTODB22</td>
<td>LUDB22</td>
<td></td>
<td>Db2</td>
</tr>
<tr>
<td>USIBMSTOSQL1</td>
<td>LUSQLDS</td>
<td>TPNSQLDS1</td>
<td>Db2 for VM production system</td>
</tr>
<tr>
<td>USIBMSTOSQL2</td>
<td>LUSQLDS</td>
<td>TPNSQLDS2</td>
<td>Db2 for VM test system</td>
</tr>
</tbody>
</table>

**Note:** "USIBMSTODB21 plans to accept requests from many Windows requesters.

**Defining the Db2 subsystem to VTAM**
You need to use an APPL definition statement and a MODEENT macro to define the Db2 subsystem to VTAM. Samples of both the APPL and MODEENT macros are in the DSN8VTAM sample data set.

**The APPL statement**
A VTAM APPL definition statement defines the VTAM options for the Db2 subsystem and includes it in a major node.

With VTAM, you can use a model application program definition for Db2. With a model definition, you use wild card characters for the application name (LU name).

Spiffy uses the statement in the table below for the USIBMSTODB21 Db2 subsystem:
For your convenience, the APPL statement example is provided in data set DSN8VTAM, in the sample library, SDSNSAMP.

The topics that follow describe the APPL options that Spiffy uses and a few more in which you might be interested. Not all APPL options are described.

Related reference:

VTAM Application Program Major Node (SNA Resource Definition Reference)

Options for which you must choose values:

For some options in your APPL statement, you must supply a specific value; for others, Db2 suggests values that are not the VTAM default values.

In your APPL statement, you must code values for the following options:

**name**

The 1 to 8 character LU name you chose. For their USIBMSTODB21 Db2 system, Spiffy uses LUDB21.

**AUTOSES**

The number of contention winner sessions that VTAM is to activate automatically between this Db2 and another system on a given mode before Db2 requests a conversation to be created.

Contention occurs when two LUs want to allocate a conversation at the same time in the same session. In order to resolve contention situations, VTAM denotes one LU as the contention winner and one as the contention loser. The winner automatically prevails and is allowed to allocate its conversation. The loser must wait to allocate its conversation.

The default is 0. The suggested value is 1 or greater to ensure that VTAM informs Db2 if a session is inactivated.

Too large a number can take up storage and create resources that are not used. A small number can result in a one-time delay to bring up additional sessions when they are needed by an application.

**DMINWNL**

The minimum number of parallel sessions in which, if there is contention for a conversation, this local Db2 subsystem is the winner.

The suggested value is one-half the value of DSESLIM, described below.
**DMINWNR**

For the same situation as described for DMINWNL, the number of sessions in which the remote system is the winner. The suggested value is one-half the value of DSESLIM, described below.

**DSESLIM**

The default maximum number of sessions allowed for this Db2 subsystem as it communicates with any other system on a given mode. For performance reasons, the Db2 suggested value for DSESLIM is the maximum number of sessions that can possibly be in use on any mode. For example, assume you have 5 modes for which the following maximum numbers of sessions could be active: 10, 12, 20, 30, 40. In this case, DSESLIM should be 40.

Because calculating a precise value for this number can be rather difficult if you do not know exactly how many applications run on a specific mode, Spiffy chooses 50. They can modify this option later if they have problems obtaining enough sessions, or if they find they are requesting sessions that they never need.

You can use DSESLIM to control the number of sessions that this subsystem can issue or receive. For example, to avoid overloading this subsystem with requests from remote application processes, you can assign a low number to DSESLIM to limit the number of simultaneous remote requests issued by a given partner and mode.

Use the CONVLIMIT column of the LUMODES table in the CDB to override this value for specific cases.

**MODETAB**

The name of the VTAM logon mode table you use to define Db2 session modes. Only modes defined in this table are eligible for conversations created by the local Db2. If you leave this blank, Db2 uses the default mode table shipped with VTAM (ISTINCLM). Spiffy decides to set up a separate mode table and chooses the name DB2MODES. Db2 cannot use either the default mode table or the one you set up yourself until you make entries into the table.

**PRTCT**

If you decided to use a password, this is that password. Later, you must store the same password in the bootstrap data set (BSDS), entering it through installation panels or the change log inventory utility.

If you prefer not to use a password, omit this option. The installation panels and the change log inventory utility do not require you to enter a password.

**SECACPT**

The level of conversation-level security allowed.

**Recommendation:** Use ALREADYV, which gives you the most flexibility in determining your security. You can use the CDB to determine levels of security on a more granular basis.

In general, SECACPT=CONV should not be used because in many cases, it does not allow already verified conversations for DRDA access.

**VERIFY**

Whether you want SNA partner LU verification. The default, VERIFY=NONE, means that any system can connect with yours. Because Spiffy is setting up a small, restricted network, it chooses the default for now.
Use VERIFY=REQUIRED to activate partner LU verification. This means that you let RACF and VTAM check the identity of an LU that is attempting to connect with yours.

Db2 has no dependency on the value you choose.

**VPACING**

The maximum number of messages that another system can send to this local Db2 subsystem during a conversation before waiting to receive a pacing response. The suggested value is 2.

These message sizes are determined by the RUSIZES option of the MODEENT macro. VPACING and RUSIZES, together with some overhead, determine the amount of storage required for the pacing window.

**Related concepts:**
- CNOS processing in Db2
- Session-level pacing
- The MODEENT macro

**Related tasks:**
- Calculating session limits
- Update SYSIBM.LUMODES with conversation limits
- Controlling LU attachments to the network (Managing Security)

**Related reference:**
- LU 6.2 Security (SNA Network Implementation Guide)

**Options that you must code exactly as given:**

In some cases, Db2 requires particular values of APPL options.

For the following options, you must code the values exactly as shown; they are not the VTAM defaults:

**APPC=YES**

Tells VTAM that Db2 uses APPC conversation verbs.

**AUTH=(ACQ)**

Determines the Db2 system authority to use certain VTAM functions.

**SRBEXIT=YES**

Tells VTAM that Db2 uses service request block (SRB) processing in its exit routines.

**SYNCLVL=SYNCPT**

Tells VTAM that Db2 supports two-phase commit. Other systems communicating with this Db2 use this indication to determine if Db2 supports the updating of many locations in one unit of work.

Coding SYNCLVL=SYNCPT does not preclude the support of partner LUs that do not support two-phase commit. Db2 still supports the non-two-phase process.

**Options that must use VTAM defaults:**

Some options in your APPL statement require you to use the VTAM default values. You do not need to code these options.

Db2 must use the VTAM default values for the following options:
HAVAIL=NO
Indicates whether an XRF session can be supported. Db2 requires the default, NO.

PARSESS=YES
Specifies that parallel sessions are allowed. This defaults to YES when APPC=YES.

ENCR=NONE
Specifies information about specific cryptographic requirements. There is no support for encryption in this release of VTAM for LU 6.2 applications; therefore, this must be NONE.

SONSCIP=NO
Specifies information about SCIP exit routines. Db2 does not have SCIP exit routines; this must be NO.

VTAMFRR=NO
Specifies whether a VTAM functional recovery routine is in effect when control is returned to Db2. Db2 uses its own recovery routines; this must be NO.

Other options of interest:

In most cases, you can reasonably use the VTAM default values for the APPL statement at first.

You can change these values later. The options are listed below in case you have some reason to not use the default values.

ACBNAME
The LU name for the Db2 subsystem. If the ACBNAME is different from the APPL name and both the originating and destination LUs are in the same VTAM domain, do not refer to the ACBNAME in a CDB definition. If the ACBNAME is not the same as the APPL name, VTAM may encounter name conflicts.

DDRAINL
Whether the local Db2 subsystem wants to accept permission to drain its allocation requests if a change-number-of-sessions (CNOS) request is received that specifies that draining is allowed. The suggested value is the default, NALLOW (do not allow draining).

DRESPLO
Whether the local Db2 is responsible for deactivating sessions when it receives a CNOS request specifying the local Db2 as the responsible system. The suggested value is the default, NALLOW (do not be responsible).

EAS
The approximate number of concurrent sessions for this Db2 subsystem. For performance reasons, it is better to estimate slightly high. The VTAM default is 509.

LMDET
The number of entries to be used for a hash table of other systems. The suggested value is the approximate number of other systems in the network. In the scenario, Spiffy decides to use the default value of 19.

MAXPVT
The maximum additional amount of private area storage that can be used by
VTAM within the DDF address space for the session-related control blocks and messages for Db2. Specifying 0 indicates an unbounded amount; this is the VTAM default.

**OPERCNO**
The ability to have a VTAM operator display and set VTAM session limits for a given LUNAME and MODENAME.

- Use ALLOW to enable a VTAM operator to change session limits dynamically without stopping DDF or changing the CONVLIMIT column of the SYSIBM.LUMODES table.
- Use NALLOW, the default, to make sure VTAM operators are not able to dynamically change session limits for Db2.

**Options that are ignored by Db2:**

Some options for VTAM APPL statements are not applicable to Db2. Do not code these options in your APPL statement.

The following options are not applicable to Db2:

<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
<th>Option 3</th>
<th>Option 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASLENT</td>
<td>ATNLOSS</td>
<td>MDTAB</td>
<td>SSCPFM</td>
</tr>
<tr>
<td>ASLTAB</td>
<td>MDLENT</td>
<td>POAQNAM</td>
<td>USSTAB</td>
</tr>
</tbody>
</table>

**The MODEENT macro**

A VTAM link between two systems is a *session*. For every session, a defined set of characteristics called a *mode* must exist in a VTAM table called a *log mode table*.

This is the table that you named in the MODETAB option of the APPL statement.

You can create your own log mode table or add mode names to the default mode table, ISTINCLM, which is shipped with VTAM. If you decide to add your modes to ISTINCLM, you can find that table in SYS1.SAMPLIB.

Spiffy uses the Db2 default modes at first, but also sets up a separate mode table for modes that are used by Db2 for distributed data processing. This table can then be populated with additional modes as they are needed.

**Related concepts:**

- The APPL statement

**Default VTAM modes:**

Use the MODEENT macro to enter each default mode into your mode table. When this table is complete, you must assemble and link-edit it into SYS1.VTAMLIB.

The following modes are the default modes:

- SNASVCMG is an optional mode. It is reserved for use by VTAM for CNOS processing and exists in the VTAM default log mode table. Because SNASVCMG is reserved for use by VTAM, do not enter it as a mode name in the CDB. If you have decided to set up a separate mode table for Db2, you can, if you choose, copy the SNASVCMG mode entry into your Db2 mode table, or just use it as it exists in the ISTINCLM mode table.
• IBMRDB is a recommended mode entry because it is used as a default for DRDA access whenever you do not explicitly assign a mode to a session. It does not exist in the default table; to use it as a default you must add it to your mode table.

Related concepts:
- CNOS processing in Db2

Related information:
- User-defined tables and data filter (SNA Resource Definition Reference)

Sample mode entries:

The sample mode entries for IBMRDB contain some options that are necessary for dependent LUs to request VTAM sessions.

These options are listed in the following table.

<table>
<thead>
<tr>
<th>COMPROT</th>
<th>PRIPROT</th>
<th>TSPROF</th>
<th>SECPROT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMFPROF</td>
<td>PSERVIC</td>
<td>TYPE</td>
<td></td>
</tr>
</tbody>
</table>

The samples in the figure below work for both dependent and independent LUs. However, if you have no dependent LUs, it is not necessary to re-assemble your existing mode table with the above options. For your convenience, a sample MODEENT is included in data set DSN8VTAM, in SDSNSAMP.

The ENCR option is ignored by LU 6.2 and is thus not included in the Db2 samples.

```
DB2MODES MODETAB
  IBMDB MODEENT LOGMODE=IBMRDB, DB2 DEFAULT MODE FOR APP-DIR ACC X
  TYPE=0, NEGOTIABLE BIND X
  SSNDPAC=X'02', SECONDARY SEND PACING COUNT X
  SRCPAC=X'00', SECONDARY RECEIVE PACING COUNT X
  RUSIZES=X'8989', RUSIZES IN-4096 OUT-4096 X
  FMFPROF=X'13', LU6.2 FM PROFILE X
  TSPROF=X'07', LU6.2 TS PROFILE X
  PRIPROT=X'B0', LU6.2 PRIMARY PROTOCOLS X
  SECPROT=X'B0', LU6.2 SECONDARY PROTOCOLS X
  COMPROT=X'50A5', LU6.2 COMMON PROTOCOLS X
  PSERVIC=X'06020000000000000122F00' LU6.2 LU TYPE
MODEEND
END
```

**Figure 130. Sample mode entries**

**MODEENT options:**

When you define your Db2 subsystem to VTAM, you can specify mode values. As you choose these values, be aware that partner systems can have different values. VTAM negotiates different values to limits that are acceptable to both systems when the session is established for the mode.

The options that are used in the MODEENT macro have the following meanings.

**name**

The name option (IBMDB2LM and IBMRDB in the examples) is optional and has no function in the specification of a logon mode table.
LOGMODE
Specifies the logon mode name that is to be used as a key for the session options in this table entry. This logon mode name corresponds to mode name columns in the CDB.

TYPE
TYPE=0 indicates that Db2 is using a negotiable BIND, which is required for communicating with dependent LUs.

SRCVPAC
Specifies the secondary receive pacing count. The Db2 suggested value is X'00'.

SSNDPAC
Specifies the secondary send pacing count. The Db2 suggested value is any nonzero number. Do not use 0; this turns off pacing, which can result in problems with IOBUF storage.

RUSIZES
Specifies the maximum length of data in bytes that can be sent and received in one request/response unit (RU). It is read as two numbers, each having two hexadecimal digits. The first number is for the send amount. The second number is for the receive amount. The suggested value of X'8989' means that VTAM sends a maximum of 4096 bytes (8 × 2⁹) across at one time, but there is no limit to how much total information can be sent.

FMPROF
This constant specifies the function management profile that is required for LU 6.2.

TSPROF
This constant specifies the transmission services profile that is required for LU 6.2.

PRIPROT
This constant specifies the primary LU protocols that are used in LU 6.2.

SECPROT
This constant specifies the secondary LU protocols that are used in LU 6.2.

COMPROT
This constant specifies the common LU protocols that are used in LU 6.2.

PSERVIC
This constant specifies this LU as an LU type 6.2.

Some of the above options can have a profound effect on performance because of their impact on pacing.

The ENCR option is not included in the sample above because it is ignored by LU 6.2.

Related concepts:
Session-level pacing

Populating the communications database
If you plan to use Db2 only as a server, you do not need to populate the CDB; default values are used. However, if you intend for Db2 to act as a requestor, you need to insert one row for each remote system into SYSIBM.LOCATIONS and SYSIBM.LUNAMES.
About this task

For example, Spiffy’s USIBMSTODB21 subsystem works as a server for many Windows requesters. It is not necessary for Spiffy to register all those requesters in the Db2 CDB.

You do not need to populate table SYSIBM.LULIST unless Db2 is acting as a requester of data that resides in a data sharing group.

After you populate these tables, you can write queries that access data at a remote system.

Procedure

- Add rows to the SYSIBM.LOCATIONS table. When you issue an SQL CONNECT statement, the LOCATION column of the SYSIBM.LOCATIONS table correlates the location name (or DRDA RDBNAM) with the VTAM LU name. If necessary, the location name is also correlated with the transaction program names (TPNs). The following SYSIBM.LOCATIONS table columns relate to VTAM:

  **DBALIAS VARCHAR(128) NOT NULL**
  The name that is associated with the server. This name is used to access a remote database server. If DBALIAS is blank, the location name is used to access the remote database server. This column does not change database object names that are sent to the remote site using a location qualifier. Use the DBALIAS column to access data at two or more different remote locations when those remote locations have the same name. The LOCATION specifies where the database is in the network, and the DBALIAS is used to access the database server. This column does not change database object names that are executed in the application using the LOCATION. All fully qualified table names must reference the server’s LOCATION name or one of its server LOCATION alias names, otherwise the SQL statement will fail because the table does not exist.

  **LOCATION CHAR(16)**
  The unique network location name, or DRDA RDBNAM, assigned to a system, remote or local. You must provide location names for any systems that you request data from. This column is the primary key for this table. If the remote LU exists in the same VTAM domain, specify the APPL name, not the ACBNAME. DBALIAS can override this name.

  **LINKNAME CHAR(8)**
  Identifies the VTAM attributes associated with this location. For each LINKNAME specified, you must have a row in SYSIBM.LUNAMES whose LUNAME matches the value specified in this column. Because this table is used for outbound requests, you must provide an LUNAME or your requests fail. Do not enter blanks in this column.

  **TPN VARCHAR(64)**
  This column is used to enter a transaction program name (TPN) for SNA conversations with non-Db2 systems. You only need to use this column if you are sending or receiving SQL requests from systems using non-default TPNs.

For example, Spiffy’s USIBMSTODB21 location wants a LOCATIONS table that looks like the following table.
### Table 140. Spiffy's LOCATIONS table

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>LINKNAME</th>
<th>TPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>USIBMSTODB21</td>
<td>LUDB21</td>
<td></td>
</tr>
<tr>
<td>USIBMSTODB22</td>
<td>LUDB22</td>
<td></td>
</tr>
<tr>
<td>USIBMSTOSQL1</td>
<td>LUSQLDS</td>
<td>TPNSQLDS2</td>
</tr>
<tr>
<td>USIBMSTOSQL2</td>
<td>LUSQLDS</td>
<td>TPNSQLDS1</td>
</tr>
</tbody>
</table>

To add the second row, Spiffy can issue the following statement:

```sql
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME)
VALUES ('USIBMSTODB22', 'LUDB22');
```

**Tip:** No row is needed for the local Db2 in the LUNAMES and LOCATIONS tables. For example, Spiffy's USIBMSTODB21 subsystem does not require a row that shows its own LU name and location name. However, for convenience, Spiffy decides to populate one LUNAMES table and one LOCATIONS table and to duplicate them entirely at each location. As a result, each table contains a row for its own LU name or location name.

- Insert rows into the SYSIBM.LUNAMES table. The LUNAMES table defines the security and mode requirements for conversations with other systems. Decisions about how to populate this table depend on how you intend to use Db2.
  - If you use this system only as a server, Db2 can use a blank in the LUNAME column as a default. Db2 uses the values in the default row as defaults for LUs that are not explicitly defined in LUNAMES. If you do not have a row with a blank in the LUNAME column, Db2 rejects client connections that do not explicitly state a valid LUNAME. The DSNTIJSG installation job creates the default row in table SYSIBM.LUNAMES.
  - If this Db2 requests data from other systems, you need to provide LU names for those systems. If the remote LU exists in the same VTAM domain, specify the APPL name, not the ACBNAME.

The LUNAMES table has the following columns:

**LUNAME CHAR(8)**

The LU name of the remote system. The default of 8 blanks indicates that this row is used for serving the requests of any system that is not specifically listed in the LUNAMES table. For example, because USIBMSTODB21 acts strictly as a server for many Windows requesters, Spiffy leaves the LUNAME column blank for those requesters and uses default values for the entire row.

However, you must provide LU names for any remote system that uses different values from the defaults.

**SYSMODENAME CHAR(8)**

This column is ignored for DRDA access conversations.

**SECURITY_IN CHAR(1)**

Defines the security options that are accepted by this Db2 subsystem when an SNA client connects to Db2. The default, A, means that an incoming connection request is accepted if it includes any of these:

- A user ID
- A user ID and password
- A user ID and RACF PassTicket
- A Kerberos security ticket.
SECURITY_OUT CHAR(1)
Defines the security option that is used when local Db2 SQL applications connect to any remote server associated with this LUNAME. The default, A, means that outgoing connection requests contain an authorization ID without a password.

ENCRYPTPSWDS CHAR(1)
For now, Spiffy uses a blank to indicate no encryption of passwords.

MODESELECT CHAR(1)
Determines whether to use the default mode or to choose a mode from the MODESELECT table. Spiffy uses a blank to use the default IBMRDB mode for conversations using DRDA access.

USERNAMES CHAR(1)
This column is used for inbound and outbound requests to control authorization ID translation.
Spiffy uses a blank to indicate that no authorization IDs are translated, and also that no passwords are sent to the server.

GENERIC CHAR(1)
A Y in this column indicates that a generic LU name is to be used for CNOS processing and SQL requests sent to the partner LU. A value of N or a blank indicates that the name specified in the LUNAME column is to be used.

For example, Spiffy’s USIBMSTODB21 location wants a LUNAMES table that looks like the following table.

Table 141. Spiffy’s SYSIBM.LUNAMES table. The row of blanks is a default row that Spiffy intends to use for Windows requesters in its initial testing.

<table>
<thead>
<tr>
<th>LUNAME</th>
<th>SYSMODENAME</th>
<th>USERSECURITY1</th>
<th>ENCRYPTPSWDS</th>
<th>MODESELECT</th>
<th>usernames</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUDB21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUDB22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUSQLDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(blanks)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note:
1. USERSECURITY refers to SECURITY_IN AND SECURITY_OUT

Spiffy can use an SQL INSERT statement to add the appropriate rows. For example, they add the LU name for USIBMSTODB22 with this statement:

```
INSERT INTO SYSIBM.LUNAMES (LUNAME) 
VALUES ('LUDB22');
```

- Add rows in the SYSIBM.USERNAMES table. The SYSIBM.USERNAMES table contains information that is needed for outbound and inbound ID translation and also for “come from” checking. For more information, see SYSIBM.USERNAMES columns (Managing Security)

Related concepts:
- Support for executing long SQL statements in a distributed environment (Db2 Application programming and SQL)
- SYSIBM.LULIST (Db2 Data Sharing Planning and Administration)
- SYSIBM.LUNAMES (Db2 Data Sharing Planning and Administration)

Related tasks:
Starting VTAM to use Db2

You do not need to code any special VTAM start options to use Db2, but you can tailor start option values for Db2 communications.

About this task

You must start VTAM before starting DDF.

Two VTAM libraries must contain definitions for Db2:
- SYS1.VTAMLST contains the definitions that define Db2 as a VTAM application.
- SYS1.VTAMLIB contains mode table definitions used by DDF. This must be an APF-authorized library, or in a concatenation of APF-authorized libraries.

Procedure

To enable VTAM to use Db2:

You can use the following VTAM command, assuming that the member DB2APPLS contains definitions for Db2:

```
NET,ACT,ID=DB2APPLS
```

Related concepts:
The MODEENT macro

Related tasks:
- Starting the Domain (SNA Network Implementation Guide)
- Defining the Db2 subsystem to VTAM

Related reference:
- VTAM Start Options (SNA Resource Definition Reference)

Tuning the VTAM system

As you begin testing with the Db2 distributed data facility, you probably need to modify VTAM options and CDB values to handle certain potential problems.

About this task

You should consult a VTAM communications expert to tune your network. Consider the following tasks when tuning VTAM for DDF.

- Controlling buffer storage
  By sending large amounts of data through the network, Db2 can cause problems with your VTAM I/O buffer pool.
- Controlling pacing
  You probably need to tune your pacing options if your VTAM buffers become overloaded with data that is sent to this local Db2.
- Modifying default session limits
Consider modifying session limits if you have problems obtaining enough sessions to handle your distributed workload efficiently.

- Modifying class of service
  Specifying a class of service can help you assign priorities to your network applications.

- Associating applications with modes
  Tuning the system can require that you add new modes to your log mode table so that there is a greater variety of classes of service available for your sessions. This variety allows you to have more flexibility in tuning the system for specific uses. This topic tells you how to associate specific sessions with modes.

Before you begin tuning the network, you must understand the relationship between VTAM options and associated values in the CDB for Db2. The table below summarizes the relationship.

**Table 142. Relationship between the CDB for Db2 and VTAM macros**

<table>
<thead>
<tr>
<th>Macro Name</th>
<th>Option</th>
<th>CDB table.column</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPL</td>
<td>name</td>
<td>LOCATIONS.LINKNAME LUNAMES.LUNAME LUMODES.LUNAME MODESELECT.LUNAME USERNAMES.LINKNAME LULIST.LINKNAME</td>
<td>The LU name used in VTAM communication. This name maps 1:1 to the system's location name in LOCATIONS.</td>
</tr>
<tr>
<td>APPL</td>
<td>DSESLIM</td>
<td>LUMODES.CONVLIMIT</td>
<td>CONVLIMIT overrides session limits specified with DSESLIM. Session limit values are used in CNOS processing.</td>
</tr>
<tr>
<td>MODEENT</td>
<td>LOGMODE</td>
<td>LUMODES.MODENAME</td>
<td>LUMODES creates session limits for specific LU name and mode name combinations.</td>
</tr>
<tr>
<td>MODEENT</td>
<td>LOGMODE</td>
<td>MODESELECT.MODENAME</td>
<td>MODESELECT maps authorization IDs and plans to specific modes.</td>
</tr>
</tbody>
</table>

Related tasks:

- [Improving performance for applications that access distributed data (Db2 Performance)](Improving%20performance%20for%20applications%20that%20access%20distributed%20data%20(Db2%20Performance))

### Controlling buffer storage

VTAM uses buffer pools for control blocks, network traffic data, and channel programs. A shortage of buffer pools can have an adverse effect on VTAM CPU time, storage consumption, and the ability to serve Db2 requests.

**Procedure**

To control buffer storage:

- Monitor VTAM buffer pools by using one of the following methods:
  - The VTAM command DISPLAY NET,BFRUSE
  - A VTAM trace, obtained by entering the following z/OS MODIFY command:
    
    ```
    F procsname,TRACE,TYPE=SMS,ID=VTAMBUF
    ```
    
    `Procsname` in the command is the VTAM start procedure name. The data is collected by the generalized trace facility (GTF).
  - Use any of the following methods to prevent shortages of these VTAM buffers (IOBUFS):
Increase the number of IOBUF buffers

The IOBUF pool definition is one of the VTAM start options. You can enter the IOBUF option from the z/OS console, or you can include it at VTAM startup in SYS1.VTAMLST in member ATCSTRxx.

Tuning the IOBUF pool encompasses both base allocation and dynamic expansion values. At installation, you can specify a base allocation for the IOBUF pool (in number of buffers) and a dynamic expansion (in number of buffers). When storage runs short in the buffer pool, VTAM temporarily expands the IOBUF pool by the dynamic expansion value, based on a trigger which you can also specify in VTAM definitions.

Recommendation: Set a maximum size for the IOBUF pool size using the xpanlim start option for the buffer pool. If you turn off pacing accidentally, xpanlim prevents Db2 from causing VTAM to grab unlimited amounts of storage.

Decrease the session level pacing count

Pacing is vital for controlling the potentially large amounts of data that are transferred around the network.

Decrease the number of concurrent conversations

You can reduce the number of concurrent conversations by reducing the number of sessions.

Decrease the request unit (RU) size

The RUSIZES option is part of the mode entry statement.

Because reducing the number of sessions and the RUSIZES value can adversely affect performance, you should first consider increasing IOBUF buffers and decreasing the session pacing count.

Related concepts:
Session-level pacing
The MODEENT macro

Related tasks:
Calculating VTAM I/O buffer pool (IOBUF) storage

Related reference:
Buffer pools (SNA Network Implementation Guide)

Session-level pacing

Session-level pacing is the mechanism by which the receiver of data (Db2, in this case) can control the pace at which the sender sends data (in the form of RUs).

The pacing size is the number of RUs that VTAM sends across the line at one time, and you can set that value by using the VPACING option of the VTAM APPL definition statement. You set the RU size in the MODEENT macro. The receiving VTAM stores these RUs in its IOBUF pool; it uses pacing so that its buffers do not become flooded with data.

The pacing process works as shown in the following figure. The system at the sending side (assume it is USIBMSTODB22) passes data to its VTAM system. VTAM formats the data into RUs and sends those RUs across the network. If, for example, the pacing size is 2, then it sends two RUs. A 29-byte network header is sent with each RU.
After the USIBMSTODB22 VTAM system sends the specified number of RUs, it does not send any more data on this session until it receives a pacing response from the VTAM system at USIBMSTODB21. The USIBMSTODB21 VTAM system does not send a response until VTAM transfers the data into the Db2 buffers.

![Diagram of pacing](image)

**Figure 131. How pacing works**

Although it is generally true that the receiving system can control inbound pacing, both communicating systems negotiate final pacing values.

**Related concepts:**
- The APPL statement
- The MODEENT macro

**Related reference:**
- [Session-level pacing (SNA Network Implementation Guide)](SNA Network Implementation Guide)

**Recommendation for APPL pacing option:**

The VPACING option of the APPL statement determines the maximum number of RUs that another LU can send, on a session, to this LU. When the maximum number of RUs is sent, the other LU waits to receive a pacing response.

Specify a nonzero value for this option unless you want to turn off all pacing for all sessions that are affected by this option.

**Recommendation:** Start with a value of 2 for both communicating systems. This pacing size is the same in both directions for all modes.

The VPACING value is used with the RUSIZES option of the MODEENT macro to control the pacing window size. Thus, if VPACING is 2 and RUSIZES is 4-KB (X'8989'), then about $2 \times 4\text{KB} = 8\text{KB}$ are sent before waiting to receive a pacing response. You should verify that your VTAM buffer pools are large enough to accommodate the chosen pacing and RU sizes.

**Related tasks:**
- [Calculating VTAM I/O buffer pool (IOBUF) storage](Calculating VTAM I/O buffer pool (IOBUF) storage)

**Recommendation for MODEENT pacing options:**

The MODEENT macro contains several pacing options.

**PSNDPAC**

This does not apply to Db2; therefore, you can ignore this option.
SSNDPAC

This option is really a flag that you set to either 0 (off) or nonzero (on). When 0, outbound pacing for sessions is disabled, which can lead to severe problems with IOBUF storage.

Recommendation: Specify a nonzero value for this option.

SRCVPAC

If 0, which is the recommended value, the VPACING option of the VTAM APPL statement controls both the send and receive pacing for all sessions in all modes. A value of 0 makes it easier for you to predict pacing results and makes it easier to maintain your pacing definitions.

If nonzero, VPACING controls pacing in one direction, and SRCVPAC controls it in another. LU 6.2 protocols make it difficult to predict which option is in control at any given time.

Related concepts:
The MODEENT macro

Modifying class of service

You can define the transmission priority and paths between systems with entries in a class of service (COS) table.

About this task

Each entry in the table is associated with a list of routes to be used with a particular class. For example, you might want to place interactive sessions on a faster route than a batch job.

Procedure

To specify a class of service for a specific mode:

Use the COS option of the MODEENT macro. When you specify a name of a COS entry in the mode description, you select the list of routes that you want to be used for the session. When VTAM establishes the session, it chooses the first available route in the list of routes that you tell it to use for that class. If you do not specify a COS name, the mode gets the default list of routes from VTAM.

Associating applications with modes

As you tune your system, you can assign certain applications, such as a high-priority job, to the mode that is best suited for that job.

About this task

You can also use a specific mode assignment for an application that uses many conversations. You can assign such an application to a mode that allows more conversations than the VTAM DSESILIM value you entered in the APPL statement.

To associate a specific mode with a particular session, you need to update or insert rows into three tables in the CDB: LUNAMES, LUMODES, and MODESELECT.
Related concepts:
- When updates to CDB values take effect for VTAM (SNA) connections

Update LUNAMES to associate modes with LU names:

The LUNAMES table associates a mode with each remote system that the local subsystem can send a query to.

About this task

You can specify the following type of connections in this table:

**SQL processing conversations**
For DRDA access, an SQL processing conversation is established. The mode name for SQL processing conversations is determined by the MODESELECT table of the CDB. If the MODESELECT column of LUNAMES table is blank or contains N, then the default mode (IBMRDB) is used. If it contains a Y, then MODESELECT is searched.

Related tasks:
- Populating the communications database for use with TCP/IP

Update SYSIBM.LUMODES with conversation limits:

Use SYSIBM.LUMODES to provide VTAM with conversation limits for specific LU name and mode name combinations.

About this task

Populating SYSIBM.LUMODES is optional; if you do not specify mode names in this table, the VTAM defaults are used. The table is unlike the DSESXLIM option of the VTAM APPL definition statement, which provides the default session limits for all LU name and mode name combinations. The primary key for this table is formed by the LU name and mode name combination. Only one entry with the same LU name and mode name is allowed.

LUMODES is accessed for negotiation of session limits with a remote Db2 for a specific mode. This negotiation is called *change number of sessions* (CNOS).

Example

For example, suppose Spiffy wants to allocate 75 sessions instead of 50 (the value in DSESXLIM) for conversations to USIBMSTODB22, using the mode named LOC2MODE. They can use the INSERT statement below to update the value in the
CONVLIMIT column to 75. The new session limit takes effect the next time DDF is started, or in the initial connection to this LU for this mode.

```
INSERT INTO SYSIBM.LUMODES VALUES ('LUDB22', 'LOC2MODE', 75, 'N');
```

CNOS processing negotiates a value that is the lesser of the number of sessions available at either system for that mode. Therefore, USIBMSTODB22 must also increase its CONVLIMIT value to at least 75.

See [CNOS processing in Db2](#) for an example of how CNOS processing works.

**Columns of the LUMODES table:**

- **LUNAME CHAR(8)**
  
  Again, this is the LU name of the other system. This column is a foreign key of the LUNAMES table; thus, all LU names defined in this table must be defined in LUNAMES. When you delete an LU name from the LUNAMES table, all associated rows in LUMODES are deleted.

- **MODENAME CHAR(8)**
  
  The name of the logon mode description in the VTAM logon mode table that VTAM uses when creating a conversation to support the request from the local Db2 for data from another system. The mode named here must exist in the mode table used by Db2 before a conversation can be created between USIBMSTODB21 and USIBMSTODB22.

- **CONVLIMIT SMALLINT**
  
  The maximum number of conversations to be concurrently active between this Db2 subsystem and the other system for this mode. This number overrides the number in the DSESLIM option of the VTAM APPL definition statement during CNOS processing.

**Related concepts:**

[CNOS processing in Db2](#)

**Update SYSIBM.MODESELECT to associate plans with modes:**

SYSIBM.MODESELECT correlates authorization IDs and plan names with mode names. The primary key for this table is the combination of AUTHID, LUNAME, and PLANNAME.

**About this task**

Only one entry with the same AUTHID, LUNAME, and PLANNAME is allowed.

Use this table to make sure that certain authorization IDs using certain plans always have a predefined class of service suited for that operation. For example, the USIBMSTODB21 location might want to work with USIBMSTODB22 to set up a high performance mode for DBADM to run queries to USIBMSTODB22. After the following statement is committed, all subsequent threads to USIBMSTODB22 use mode DB2MODE1 to process SQL processing conversations:

```
INSERT INTO SYSIBM.MODESELECT VALUES ('DBADM', '', 'LUDB22', 'DB2MODE1');
```
Populating this table is optional. If the remaining columns are blank for any given LU name, then the mode name applies to all authorization IDs for all PLANNAMES accessing the given LU name.

Columns of the MODESELECT table:

AUTHID CHAR(8)

The authorization ID of the request for data from another system. A blank AUTHID indicates that the specified mode name applies to all authorization IDs. Blank is the default.

PLNAME CHAR(8)

The plan name associated with the request for data from another system. A blank plan name indicates that the specified mode name applies to all plan names. Blank is the default.

LUNAME CHAR(8)

The LU name to which the specific mode name applies. This column is a foreign key of the LUNAMES table; therefore, all LU names defined in this table must be defined in LUNAMES.

MODENAME CHAR(8)

The name of the logon mode description in the VTAM logon mode table that is used when creating a conversation to support the request for data from another system. If this column is blank, the default mode (IBMDB2LM or IBMRDB) is used.

How an SQL processing conversation mode is chosen:

The MODESELECT table of the CDB is used to choose a mode for an SQL processing conversation (if the MODESELECT column of the LUNAMES table contains Y for this LU name). The table below shows the search order of the MODESELECT table.

Table 143. Precedence search order for MODESELECT table of CDB

<table>
<thead>
<tr>
<th>AUTHID</th>
<th>PLNAME</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>The MODENAME applies to the named AUTHID for the named PLNAME accessing the named LU.</td>
</tr>
<tr>
<td>Name</td>
<td>Blank</td>
<td>The MODENAME applies to the named AUTHID for all PLNAMEs accessing the named LU.</td>
</tr>
<tr>
<td>Blank</td>
<td>Name</td>
<td>The MODENAME applies to all AUTHIDs for the named PLNAME accessing the named LU.</td>
</tr>
<tr>
<td>Blank</td>
<td>Blank</td>
<td>The MODENAME applies to all AUTHIDs for all PLNAMEs accessing the named LU.</td>
</tr>
</tbody>
</table>

If the MODESELECT column of the LUNAMES table contains Y for a particular LU name and no row is found for that LU name in the MODESELECT table, then you receive a negative SQL return code when trying to access the system at that LU.

Plan name for remote bind operations: If you want to specify a particular mode for remote bind operations, use the plan name DSNBIND in MODESELECT.
When updates to CDB values take effect for VTAM (SNA) connections

Any table in the CDB can be updated while DDF is active.

The changes take effect as follows:

- Updates to LUMODES take effect the next time DDF is started, or on the initial session to a given LUMODE combination.
- Updates to LUNAMES, LOCATIONS, and LULIST take effect as follows:
  - If DDF has not yet tried to communicate with a particular remote location, rows added to LUNAMES and LOCATIONS take effect when DDF attempts to communicate with that location.
  - If DDF has already attempted communication with a particular location, rows added to LUNAMES and LOCATIONS take effect the next time DDF is started.
- Updates to USERNAMES and MODESELECT take effect at the next thread access.
- Deletion of rows from any CDB table takes effect the next time DDF is started.

In all cases, existing conversations continue to operate as the table specified before the update.

Recommendation: If you are switching from TCP/IP to VTAM connections, stop and start DDF after you modify the CDB tables.

Important: The process of modifying the CDB, particularly MODESELECT and USERNAMES, can interfere with DDF access to the tables. This could potentially cause deadlocks and timeouts, which cause the attempted access to the remote system to fail.

Calculating session limits

You can calculate session limit values for DRDA access.

About this task

You might have to derive a precise figure for your session limits (DSESILIM). For example, if you specify a very large number for your session limits and you are running short of space, it might help to calculate a number closer to what you actually need.

Procedure

To calculate the session limit for a specific mode:

For DRDA access, determine the maximum number of concurrently active applications that use the mode to and from the remote system.

Example

For example, suppose location A has a maximum of 3 DRDA access applications that can run concurrently on Model1 to other locations. Also suppose that a maximum of 10 DRDA access applications can be incoming to location A on Model1. In this case, Model1 should support 13 sessions for DRDA access applications.
Calculating VTAM I/O buffer pool (IOBUF) storage
You can estimate VTAM buffer pool storage when you use the distributed data facility of Db2. Every path information unit (PIU) that enters or leaves VTAM resides in one or more IOBUF buffers.

About this task
A PIU is composed of a 26-byte transmission header, a 3-byte request/response header, and the request/response unit (RU) that contains VTAM application data. You define the length of the RU in a mode entry, using the RUSIZES option.

Procedure
To calculate the maximum number of buffers that are required for the local Db2 subsystem:
1. Calculate the number of buffers that each PIU occupies, and call it PIUBUF.
   
   \[ PIUBUF = \text{CEILING}\left(\frac{29 + \text{RUSIZE}}{\text{BUFSIZE}}\right) \]
   
   RUSIZE is the length of the RU in bytes. It is assumed to be the same for both session directions. BUFSIZE is the value you specified in the IOBUF pool definition.
   
   Assume you have a buffer size of 441 bytes, and an RUSIZE of 4096. With these values, PIUBUF would be 10 ((29+4096) / 441, rounded up).
   
   For channel-to-channel (CTC) and NCP connections, you need to be concerned with the VTAM MAXBFRU value. For CTC connections, MAXBFRU is the number of 4KB buffers allocated to hold the PIUs sent over the channel. If your RU size is 4096 and you allow 29 bytes for the header, then you need to allocate at least 2 4KB buffers. Thus, you need a MAXBFRU value of at least 2. When you route data through NCP, MAXBFRU is the number of VTAM IOBUF buffers allocated to hold the PIUs sent to the NCP, which means MAXBFRU must be at least as large as PIUBUF.

2. Calculate the maximum number of IOBUF buffers used by a session, and call it SESSBUF.
   
   \[ SESSBUF = \text{PACECNT} \times \text{PIUBUF} \]
   
   PACECNT stands for pacing count. For this example, assume that pacing is the same in both directions, and it is the same for all modes.
   
   If pacing is set to 2, then SESSBUF is 20.

3. Calculate the maximum number of sessions that can be active for all modes to all systems and call it SESCNT. Calculate the maximum for each mode, then add those results to get SESCNT.

4. Calculate the maximum number of VTAM buffers used by Db2, and call this DB2BUF. The formula for DB2BUF is based on a worst case scenario, because it assumes that all sessions are used by concurrent conversations.
   
   \[ DB2BUF = \text{SESCNT} \times \text{SESSBUF} \]
   
   If you assume that the maximum number of sessions that can be active is 50 (SESCNT), then 1000 is the number of IOBUF entries required by Db2 in a worst case scenario.

5. Calculate actual VTAM buffer storage consumption used by Db2, and call it STORAGE.
   
   \[ \text{STORAGE} = \text{DB2BUF} \times (\text{BUFSIZE} + 71) \]
Each buffer includes 71 bytes for VTAM internal headers.

So, to continue the above example, an upper value of real storage is estimated as follows:

\[
1000 \times (441 + 71) = 500\text{KB}
\]

**Related concepts:**
- Session-level pacing
- Definitions for channel-connected Db2 subsystems
- NCP-connected Db2 subsystems

**Related tasks:**
- Calculating session limits

**CNOS processing in Db2**

The distributed data facility of Db2 can request to alter the number of sessions with another system for a specific VTAM logon mode. This automatic process is called change number of sessions (CNOS).

This topic contains a brief overview of the process as it relates to Db2; it should help you understand the messages that CNOS processing generates.

**When sessions are started:** The AUTOSES option of the VTAM APPL determines whether, and how many, sessions are started at the time CNOS is negotiated. If AUTOSES is 0, then the sessions are not started at CNOS negotiation time; they are started as they are needed. AUTOSES should generally not be 0, because then Db2 is not informed if CNOS fails, and you receive a “resource unavailable” SQL code with the first SQL request to the remote system.

If AUTOSES is not 0, then sessions are started as follows:

- If AUTOSES is equal to or less than the number of contention winner sessions for a specific Db2 subsystem, then the number of sessions that are automatically started at CNOS negotiation is equal to AUTOSES.
- If AUTOSES is greater than the number of contention winner sessions for a specific Db2 subsystem, only the contention winner sessions are automatically started at CNOS negotiation.

Each LU has its own value for the number of contention winner sessions to start. The total number of sessions started on behalf of a CNOS negotiation request is the sum of the sessions started at each site.

**Example:** Suppose the Db2 subsystems at USIBMSTODB21 and USIBMSTODB22 have the following values in their VTAM APPL statements and LUMODES tables:
Assume that USIBMSTODB21’s DDF is started first. CNOS processing fails because USIBMSTODB22’s DDF has not yet started, and you get a message at the console. When USIBMSTODB22’s DDF is started, CNOS processing can begin. The CNOS negotiation process is as follows:

1. USIBMSTODB21 has a DSESLIM value of 50, and a CONVLIMIT value of 80. For this first step in the CNOS negotiation, the CONVLIMIT value overrides the DSESLIM value, so USIBMSTODB21 sends an initial CNOS value of 80 to USIBMSTODB22. During this initial negotiation step, the LUMODES information for USIBMSTODB22 is not yet used. USIBMSTODB22 replies to USIBMSTODB21 with its DSESLIM value of 40, which is less than the initial CNOS value of 80. An initial negotiated value of 0 to 40 sessions is therefore established. A second and final negotiation, as described in step 2, is then initiated by USIBMSTODB22.

2. Everything up to this point occurred because USIBMSTODB21 initiated CNOS, resulting in an initial negotiation of 0 to 40 sessions. Now, as shown in the following figure, USIBMSTODB22 initiates a second CNOS negotiation back to USIBMSTODB21 because USIBMSTODB22’s LUMODES table has a CNOS limit specified by the CONVLIMIT value of 50. USIBMSTODB21’s VTAM sees that Db2 allows up to 80 sessions, so VTAM sends the CNOS reply message back to USIBMSTODB22 with an unchanged value of 50. USIBMSTODB22’s CONVLIMIT value of 50 is compared with USIBMSTODB21’s overridden value of 80 from the previous CNOS, and a final value of 40 to 50 sessions is

---

### Figure 132. CNOS negotiation example: VTAM and Db2 definitions

Assume that USIBMSTODB21’s DDF is started first. CNOS processing fails because USIBMSTODB22’s DDF has not yet started, and you get a message at the console. When USIBMSTODB22’s DDF is started, CNOS processing can begin. The CNOS negotiation process is as follows:

1. USIBMSTODB21 has a DSESLIM value of 50, and a CONVLIMIT value of 80. For this first step in the CNOS negotiation, the CONVLIMIT value overrides the DSESLIM value, so USIBMSTODB21 sends an initial CNOS value of 80 to USIBMSTODB22. During this initial negotiation step, the LUMODES information for USIBMSTODB22 is not yet used. USIBMSTODB22 replies to USIBMSTODB21 with its DSESLIM value of 40, which is less than the initial CNOS value of 80. An initial negotiated value of 0 to 40 sessions is therefore established. A second and final negotiation, as described in step 2, is then initiated by USIBMSTODB22.

### Figure 133. Result of CNOS negotiation started by USIBMSTODB21. Overridden values are noted with asterisks (*).

After the number of sessions is negotiated, both systems begin starting the number of sessions that are specified in their respective AUTOSES options. VTAM does not start all 40 sessions unless the two AUTOSES values total up to 40 or greater. Instead, VTAM delays starting the other sessions until they are needed.

2. Everything up to this point occurred because USIBMSTODB21 initiated CNOS, resulting in an initial negotiation of 0 to 40 sessions. Now, as shown in the following figure, USIBMSTODB22 initiates a second CNOS negotiation back to USIBMSTODB21 because USIBMSTODB22’s LUMODES table has a CNOS limit specified by the CONVLIMIT value of 50. USIBMSTODB21’s VTAM sees that Db2 allows up to 80 sessions, so VTAM sends the CNOS reply message back to USIBMSTODB22 with an unchanged value of 50. USIBMSTODB22’s CONVLIMIT value of 50 is compared with USIBMSTODB21’s overridden value of 80 from the previous CNOS, and a final value of 40 to 50 sessions is
If the new negotiated value is smaller than the number already started by 
USIBMSTODB21, VTAM terminates the number of sessions that makes up 
the difference. If the CONVLIMIT value at USIBMSTODB22 is 20, for example, VTAM 
terminates 20 sessions on behalf of the request from USIBMSTODB22 because the 
lowest negotiated value always wins. If a session is currently being used by a 
conversation, the session is terminated as soon as the conversation is deallocated.

**Related reference:**
Managing Sessions ([z/OS Communications Server: SNA Programmer’s LU 6.2 Guide](#))

### Sample VTAM definitions to connect two Db2 subsystems

Sample VTAM definitions are provided to give you some guidance on setting up 
your network to connect two Db2 subsystems. The samples are not intended to 
give you information about all the options.

This set of definitions includes the basic definitions you need to connect two Db2 
subsystems. Additional options for channel-to-channel and Network Control 
Program (NCP) connections are covered as well.

**Basic VTAM definitions**
Some basic definitions are required for all VTAM connections.

These definitions are listed below.

```
* *******************************************************
** APPL STATEMENT FOR SYSTEM 1                           *
* *******************************************************
DB1APPL       APL=APPC=YES, X
             ATNLOSS=ALL, X
             AUTH=(ACQ), X
             AUTOSES=1, X
             DSESLIM=20, X
             DMINWNL=10, X
             DMINWR=10, X
             PRCT=DB1PWD, X
             SECACPT=ALREADYV, X
             EAS=509, X
             MODETAB=DB2MODES, X
             PARSESS=YES, X
             SRBEXIT=YES, X
             SYNCLVL=SYNCPT, X
             VPACING=2
```
** APPL STATEMENT FOR SYSTEM 2 **

DB2APPL  APPL APPC=YES,
ATNLOSS=ALL,
AUTH=(ACQ),
AUTOSES=1,
DSESLLM=20,
DMINWNL=10,
DMINWNR=10,
PROCT=DB2PWD,
SECACPT=ALREADYV,
EAS=509,
MODETAB=DB2MODES,
PARSESS=YES,
SRBEXIT=YES,
SYNCLVL=SYNCPCT,
VSPACING=2

** LOGMODE TABLE FOR SYSTEM 1 TO SYSTEM 2 CONNECTIONS **

DB2MODES MODETAB

*  LU6.2 SERVICES MANAGER BASE MODE FOR APPC/VTAM

SNASVCMG MODEENT LOGMODE=SNASVCMG,
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'80',
SECPROT=X'80',
PSERVIC=X'060200000000000000000300',
COMPROT=X'0801',
RUSIZES=X'8585',
ENCR=B'0000'

** DB2 DEFAULT FOR SYSTEM 1 AND SYSTEM 2 DRDA ACCESS **

IBMDB MODEENT LOGMODE=IBMDB,
TYPE=0,
PSNDPAC=X'00',
SSNDPAC=X'02',
SRCVPAC=X'00',
RUSIZES=X'8989',
FMPROF=X'13',
TSPROF=X'07',
PRIPROT=X'80',
SECPROT=X'80',
PSERVIC=X'060200000000000000000300',
CMPOUT=X'50A5',
CMPPRT=X'50A5'

** ATCSTRTA VTAM START OPTIONS FOR SYSTEM 1, INCLUDES IOBUF **

CONFIG=TA,
SSCPID=53, MAXSUBA=150, HOSTSA=53,
SSCPNAME=SSCP004, NETID=USIBMSY,
IOBUF=(328,441,20,,64,48,768)

** ATCSTRTB VTAM START OPTIONS FOR SYSTEM 2, INCLUDES IOBUF **

CONFIG=TB,
SSCPID=54, MAXSUBA=150, HOSTSA=54,
SSCPNAME=SSCP00E, NETID=USIBMSY,
IOBUF=(328,441,20,,64,48,768)
Definitions for channel-connected Db2 subsystems

When determining your channel-to-channel definitions, remember that MAXBFRU must be large enough to handle the largest PIU.

Because Db2 is sending 4096 bytes, you need enough 4KB buffers to accept 4096 + 29 bytes (the 29 bytes is for the network header). Thus MAXBFRU must be at least 2 in our example.

In many cases, the Db2 RU size is larger than any other PIUs used on existing CTCs, which can mean you must examine your MAXBFRU values on existing CTC definitions. If the values are too small, you get an SNA X’800A’ sense code, indicating that the PIU was truncated during transmission.
*** CTC DEFINITIONS FOR SYSTEM 1 *

DB1CTC VBUILD TYPE=CA CTC MAJOR NODE DEFINITION
DB1GRP B GROUP LNCTL=CTCA, CTC LINE TYPE
MIH=YES,REPLYTO=10.0
DB1CTCL LINE ADDRESS=(500), CTC ADDRESS FOR THIS LINE
DELAY=0, CTC DELAY
MAXBFRU=8, MAX BUFFER USED
ISTATUS=ACTIVE INITIAL STATUS IS ACTIVE
DB1CTCP PU ISTATUS=ACTIVE

*** CTC DEFINITIONS FOR SYSTEM 2 *

DB2CTC VBUILD TYPE=CA CTC MAJOR NODE DEFINITION
DB2GRP B GROUP LNCTL=CTCA, CTC LINE TYPE
MIH=YES,REPLYTO=10.0
DB2CTCL LINE ADDRESS=(500), CTC ADDRESS FOR THIS LINE
DELAY=0, CTC DELAY
MAXBFRU=8, MAX BUFFER USED
ISTATUS=ACTIVE INITIAL STATUS IS ACTIVE
DB2CTCP PU ISTATUS=ACTIVE

*** PATH - NETWORK ROUTES FOR SYSTEM 1 *

MVSDB2 PATH DESTSA=2,ER1=(2,1),VR1=1,
VRPWS10=(2,30),VRPWS11=(2,30),VRPWS12=(2,30)

*** PATH - NETWORK ROUTES FOR SYSTEM 2 *

MVSDB1 PATH DESTSA=1,ER1=(1,1),VR1=1,
VRPWS10=(2,30),VRPWS11=(2,30),VRPWS12=(2,30)

*** CDRSC DEFINITIONS FOR SYSTEM 1 *

VBUILD TYPE=CDRSC
DB2APPL CDRSC CDRM=DB2CDRM,ISTATUS=ACTIVE

*** CDRSC DEFINITIONS FOR SYSTEM 2 *

VBUILD TYPE=CDRSC
DB1APPL CDRSC CDRM=DB1CDRM,ISTATUS=ACTIVE

*** CDRM DEFINITIONS FOR SYSTEM 1 AND 2 (SAME DEFINITION USED) *

VBUILD TYPE=CDRM
DB1CDRM CDRM SUBAREA=1,ISTATUS=ACTIVE,CDRSC=OPT
DB2CDRM CDRM SUBAREA=2,ISTATUS=ACTIVE,CDRSC=OPT

*** ATCCONTA - NETWORK CONFIGURATION LIST FOR SYSTEM 1 *

DB1PATH,DB1CTC,DB1RSC,DB1APPLS,DB1DRMS

*** ATCCONTB - NETWORK CONFIGURATION LIST FOR SYSTEM 2 *

DB2PATH,DB2CTC,DB2RSC,DB2APPLS,DB2DRMS

Figure 136. Channel-to-channel (CTC) definitions
NCP-connected Db2 subsystems

The Advanced Communications Facility/Network Control Program (ACF/NCP) is a product that you can use to generate a network control program load module. The network control program load module is loaded from the host into a communications controller.

The network control program controls the lines and devices attached to it. It transfers data to and from the devices and handles any errors that occur, including retries after line errors.

A communications controller can be locally attached to a host via a channel, or it can be link-attached to another communications controller that is channel-attached.

Our sample definitions are used for the following setup:

```
  +----------+  +----------+
  | SYSTEM1  |  | NCP      |
  +----------+  +----------+
  |  PCCU     |  |  PCCU     |
  +----------+  +----------+

When you are defining your NCP connections, remember the following:

- MAXBFRU must be large enough to handle the biggest PIU that is sent to the NCP. In our example, Db2 is sending 4125 bytes per PIU (4096 + a 29-byte network header). Given an IOBUF buffer size of 441 bytes, MAXBFRU must therefore be at least 10 (10 × 441 = 4410, which is greater than 4125).
- The MAXDATA option must also be large enough to handle biggest PIU (RUSIZE + 29 bytes).

If Db2 is using existing NCP definitions, you should make sure your MAXBFRU and MAXDATA options are large enough. If these values are too small, you get an SNA X’800A’ sense code, indicating that the PIU was truncated during transmission.
```
DUMPDS=DUMPDS, DUMP DATA SET X  
CDUMPDS=CDUMPDS, CSP DUMP DATA SET X  
MDUMPDS=MDUMPDS, MOSS DUMP DATA SET X  
INITEST=NO, NO 3745 INITIAL TESTS AT LOAD TIME X  
MAXDATA=4302, = BFRS*TRANSFR - 18 X  
OWNER=HOST2, X  
SUBAREA=4, HOST SUBAREA X  
VFYLM=YES  
***********************************************************************  
* BUILD MACRO SPECIFICATIONS *  
***********************************************************************  
ACFNCPBD BUILD BFRS=(240), NCP BUFFER SIZE, # EP BUFFERS X  
BRANCH=1000, X  
CATRACE=(YES,10), X  
CSMHDR=27FC711C3F0405C40C8C4D9405C, X  
CSMHDRC=40E3C5E7E3405C5C, X  
CSMSG=5C5C40E5E3C1D440E2C8E4E3C4D6E6D540, X  
CSMSGC=6040C8C1E240C2C5C7E4D5405C5C, X  
DIALTO=60, WAIT 1 MIN FOR AUTOCALL ANSWER X  
DR3270=NO, NO DYNAMIC RECONFIG X  
DSABLTO=3.0, TIME TO DETECT DSR DROP X  
ENABLE=2.2, TIME TO DETECT DSR AFTER ENABLE X  
LOADLIB=NCPLOAD, LIBRARY FOR ACF/NCP LOAD MODULE X  
LTRACE=8, UP TO 8 LINES CONCURRENTLY TRACED X  
MAXSSCP=8, NUMBER OF SSCPS IN SESSION X  
MBSIZE=4M, AMOUNT OF MEMORY X  
MODEL=3745, 3745 MODEL 410 X  
NETID=BCR1, 3745 MODEL 410 X  
NEWNAME=DDBLC0, LOAD MODULE NAME X  
PUNAME=DDB, X  
PUNAME=DDB, X  
NPA=YES, NPA WILL NOT COLLECT DATA X  
OLT=NO, INCLUDE ONLINE TEST FACILITY-OLTEP X  
PRTGEN=NOGEN, DON'T PRINT ASSEMBLED STATEMENTS X  
PWROFF=NO, X  
SLODOWN=15, SLOWDOWN AFTER 15% BUFFERS AVAIL X  
SUBAREA=26, NCP SUBAREA X  
TRACE=(YES,10), 10-16 BYTE ADDRESS TRACE ENTRIES X  
TRANSFR=18, =(4096+51)/BFRS--ROUNDED UP X  
TRCPIL=2000, SIZE OF LINE AND SIT TRACE X  
TYPGEN=NCP, X  
TYPHTS=MVS, MVS OPERATING SYSTEM X  
USGTIER=5, NCP USAGE TIER - REQUIRED X  
VERSION=V5R3, X  
XBREAK=NONE  
***********************************************************************  
** SYSCNTRL OPTIONS - REQUIRED BY VTAM **  
**  
** SYSCNTRL OPTIONS=(ENDCALL, MODE, RCNTRL, RCOND, RCEMD, RIMM, X  
SESSION, NAKLIM, LNSTAT, SSPAUSE, XMTLMT, BHASSC, STORDSP) **  
***********************************************************************  
HOST2 HOST BFRRD=0, VTAM REQUIREMENT FOR OS X  
INBFRS=18, INITIAL BUFFERS FOR EACH RECEIVE X  
MAXBFU=10, < BASENO IN IOBUF FOR VTAM X  
SUBAREA=4, X  
UNITSZ=441 = BUFSIZE IN IOBUF FOR VTAM X  
HOST1 HOST BFRRD=0, VTAM REQUIREMENT FOR OS X  
INBFRS=18, INITIAL BUFFERS FOR EACH RECEIVE X  
MAXBFU=10, < BASENO IN IOBUF FOR VTAM X  
SUBAREA=4, X  
UNITSZ=441 = BUFSIZE IN IOBUF FOR VTAM X  

Chapter 11. Working with additional capabilities for Db2 915
Using the change log inventory utility to update the BSDS

The DDF statement of the change log inventory utility can be used to insert or update information that is stored in the bootstrap data set.

Procedure

To add or update information in the bootstrap data set:

Use the options of the DDF statement of the change log inventory utility as follows.
To update any value, you need to use only the option for that value. To insert new values, you need values for LOCATION, LUNAME, and any other values that you want to specify.

Value Option for inserting or updating
Location alias
  ALIAS names
  This value is optional. You can add or delete aliases by respecifying the ALIAS names. The new list of names replaces the existing list.
Location name
  LOCATION=name
LU name
  LUNAME=name
Generic LU name
  GENERIC=name
Password
  PASSWORD=password
  This value is optional, depending on whether you entered a password in the VTAM APPL statement
TCP/IP connection port
  PORT=number
  This value is optional. If you specify a value, it must be nonzero and different from the value of RESPORT. You can delete the port by entering 0.
TCP/IP resync port
  RESPORT=number
  This value is optional. If you specify a value, it must be nonzero and different from the value of PORT. You can delete the port by entering 0.

- To delete either a generic LU name, a password, or an alias, use one of these keywords:
  Value Statement for deleting
  Generic LU name
    NGENERIC
  Password
    NOPASSWD
  Alias
    NOALIAS

Related reference:

[DSNJU003 (change log inventory) (Db2 Utilities)]

Connecting systems with TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is a standard communication protocol for network communications. Previous versions of Db2 supported TCP/IP requesters, although additional software and configuration was required.

About this task

The domain name (IP address) and port number (service name) uniquely identify a Db2 subsystem in the TCP/IP network. The domain name and the server port number (service name) of the database server must be defined in the communications database (CDB) at a requesting Db2 so that the Db2 subsystem can connect to a remote location. If you use a port number in the CDB to access a remote Db2 location, the port number must be defined to TCP/IP. The domain
name must be defined to the TCP/IP host so that a Db2 subsystem can accept connections from remote locations. Port numbers are never used by a Db2 subsystem that accepts connections. Optionally, you can protect the port number that Db2 uses when acting as a server within TCP/IP from being used by any other task or job in the subsystem. When DDF is started, the Db2 subsystem binds itself to its designed port.

TCP/IP terminology

Db2 uses some terms that are specific to communications.

These terms are defined as follows.

IP address
Uniquely identifies a host within the TCP/IP network. This is sometimes called an internet address. A Db2 subsystem resides on a TCP/IP host. The IP address can take two forms. The first form is a 32 bit address that displays in dotted decimal format where X'05041020' displays as 5.4.16.32. The second form is a 128 bit address that displays in colon-hexadecimal format, such as FEDC:BA98:7654:3210:6543:2109:

Domain name
The fully qualified name that identifies an IP address. This can be used instead of the IP address. An example of a domain name is stlmvs1.stl.ibm.com. Some software refers to stlmvs1 as the host name and stl.ibm.com as the domain name. Db2 allows the network administrator to identify a host using a domain name.

Domain name server (DNS)
Manages a distributed directory of domain names and related IP addresses. Domain names can be translated into IP addresses and you can find a domain name associated with a given IP address. Db2 uses the gethostbyname service to get a list of IP addresses for a given domain name.

Port
Identifies an application executing in a host. For example, a port number identifies a Db2 subsystem to TCP/IP. A port number is a two byte integer value that is displayed in decimal format. This number identifies the application within a TCP/IP instance. A port number of X'01D2' displays as 466. There are three basic kinds of TCP/IP ports:

Well-known port
This is a port number between 1 and 1023 that is reserved in the TCP/IP architecture for a specific TCP/IP application. Some typical well-known port numbers are:
- FTP is port number 21
- Telnet is port number 23
- DRDA relational database is port number 446

Ephemeral port
Port numbers that are dynamically assigned to a client process by the client's TCP/IP instance. Db2 uses an ephemeral port when it is acting as the DRDA requester. This ephemeral port is associated with the requester for the life of the thread or connection.

Server port
Port numbers that are used when a TCP/IP program does not have a well-known port number, or another instance of the server program is already installed using the well-known port number. As a requestor, Db2 defaults to using the DRDA relational database well-known port number to connect to a server location. We suggest that your Db2 subsystem be defined with the DRDA
well-known port number of 446. However, the network administrator can assign a server port to the Db2 subsystem. If two different Db2 subsystems reside on the same host, acting as two different locations (a non-data-sharing group), each Db2 subsystem must have a unique port. In this case, only one Db2 subsystem can use the DRDA well-known port number.

Service name
Another way to refer to a port number. A network administrator can assign a service name for a remote location instead of using the port number.

Enabling TCP/IP communication
To enable TCP/IP communication between DRDA partners and Db2, you must complete some tasks such as preparing the Language Environment run time library.

About this task
You do not have to complete the following steps in exact order, but steps 1, 2, and 3 should be completed before the other steps.

1. Prepare the Language Environment run time library
2. Enable DDF for UNIX System Services
3. Define the Db2 subsystem to TCP/IP
4. Populate the communications database
5. Start TCP/IP support
6. Tune TCP/IP

Db2 obtains more information from the TCP/IP stack, therefore requiring more configuration than many other daemons.

If you do not plan to communicate with remote sites with SNA/APPC, you do not need to define VTAM to Db2 if you update the BSDS DDF record with an IPNAME value. When you update the BSDS DDF record with an IPNAME value, DDF does not activate its SNA/APPC communications support. Db2 TCP/IP communications uses the IPNAME value and a character representation of the TCP/IP resync port (RESPORT) hexadecimal value to identify units of work. If the BSDS DDF record does not contain an IPNAME value, you must define VTAM to Db2 because Db2 TCP/IP communications uses the VTAM network ID and LUNAME to identify units of work.

DDF enhancements enable TCP/IP communication with DRDA partners with DRDA Level 3 support and earlier. To use the new functions, clients must have the updated versions of Db2 Connect or any DRDA requester or server that supports the latest DRDA database protocols. TCP/IP connectivity lets you connect DDF to clients on multiple platforms directly.

Attention: TCP/IP is the recommended communication protocol when accessing remote systems.

Related concepts:

TCP/IP access methods (Db2 Data Sharing Planning and Administration)

Related tasks:

Connecting systems with VTAM

Related reference:
Preparing the Language Environment run time library
Because DDF uses some functions in the Language Environment library, DDF needs access to the run time library.

Procedure
To provide DDF with access to the run time library:

Complete one of the following options:

- Concatenate the Language Environment library in the z/OS link list, which does not require the library to be APF-authorized. This is the standard method. If you choose this method, remove the Language Environment library concatenation from the DDF JCL procedure.
- Include the Language Environment library in a STEPLIB concatenation for the DDF JCL procedure. The Language Environment library must be APF-authorized to be added to the DDF JCL procedure. The Db2 installation automatically adds the library to the DDF STEPLIB concatenation.

Enabling DDF for UNIX System Services and TCP/IP
DDF uses the assembler callable interface of UNIX System Services to perform TCP/IP services. Any address space that needs to use UNIX System Services must have a z/OS user ID that is defined with an OMVS segment. The address space can also have a z/OS group name.

About this task
During initialization, DDF calls UNIX System Services to obtain the maximum number of socket descriptors that can be opened concurrently. The value is defined by the MAXFILEPROC parameter in the active z/OS UNIX System Services member (BPXPRMnnn) of the z/OS PARMLIB. If the defined value is less than 131072, DDF invokes UNIX System Services to raise it to 131072, which requires the DDF user ID to have superuser authority. DDF executes as an authorized program and is protected against unauthorized use of this privilege.

The standard way to assign a z/OS user ID and a z/OS group name to a started address space is to use the z/OS Security Server (RACF) STARTED resource class. This method enables you to dynamically assign a z/OS user ID by using commands instead of requiring an IPL to have the assignment take effect. The alternative method to assign a z/OS user ID and a z/OS group name to a started address space is to change the RACF started procedures table, ICHRIN03.

Procedure
To enable DDF to use UNIX System Services:

1. Use either of the following z/OS Security Server (RACF) commands to assign an OMVS segment to a z/OS user ID:

   ADDUSER  ddffuid OMVS(UID(nnn))
   ALTUSER ddffuid OMVS(UID(nnn))

   where ddffuid is the z/OS user ID and nnn is any valid, unique identifier. If you set nnn to 0, the process has UNIX System Services superuser authorization.
You should either set the MAXFILEPROC parameter to at least 131072 or grant superuser authority to the DDF user ID. Because setting the MAXFILEPROC parameter within the active BPXPRM\nnn z/OS PARMLIB member to 131072 affects the entire system, any UNIX System Services process can open 131072 files or sockets concurrently.

2. If you do not want to set the system-wide MAXFILEPROC parameter to 131072 and do not want to grant superuser authority to the DDF user ID, you can explicitly authorize the DDF user ID to raise the limit to 131072 by using one of the following RACF commands:

   ADDUSER ddfuid OMVS(UID(nnn) FILEPROCMAX(131072))...
   ALTUSER ddfuid OMVS(UID(nnn) FILEPROCMAX(131072))...

   where ddfuid is the z/OS user ID and nnn is any valid, unique identifier.

3. If you also want to assign a z/OS group name to the address space, assign an OMVS segment to the z/OS group name by using one of the following RACF commands:

   ADDGROUP ddfgnm OMVS(GID(nnn))...
   ALTGROUP ddfgnm OMVS(GID(nnn))...

   where ddfgnm is the z/OS group name and nnn is any valid, unique identifier.

4. If you actively administer the STARTED resource class, issue the following RACF commands to assign a z/OS user ID and a z/OS group name to the DDF started procedure. DDF requires only a z/OS user ID. A z/OS group name is optional.

   RDEFINE STARTED (VC1ADIST.*) STDATA(USER(ddfuid) GROUP(ddfgnm)) ... 
   RALTER STARTED (VC1ADIST.*) STDATA(USER(ddfuid) GROUP(ddfgnm)) ...

   where VC1ADIST.* is the DDF started procedure, ddfuid is the z/OS user ID, and ddfgnm is the z/OS group name.

**Requirement:** The profile name that you specify in the RACF command must be in the generic format; that is, the profile name must end with a period followed by an asterisk (*).

5. Issue the following RACF command so that the changed profile takes effect:

   SETROPTS RACLIST(STARTED) REFRESH

6. When a member-specific dynamic VIPA (VIPARANGE DVIPA) is specified in the Db2 BSDS using the DSNJU003 keywords IPv4 or IPv6, DDF internally activates the DVIPA using the SIOCSVIPA/SIOCSVIPA6 IOCTL. However, you must either grant superuser authority to the DDF user ID or define a security profile for MODDVIPA under the SERVAUTH class to authorize the DDF user ID to execute the SIOCSVIPA/SIOCSVIPA6 IOCTL. You can define the security profile using RACF as follows:

   RDEFINE SERVAUTH (EZB.MODDVIPA.sysname.tcpname)UACC(NONE) 
   PERMIT EZB.MODDVIPA.sysname.tcpname ACCESS(READ) CLASS(SERVAUTH) ID(USER1)

   • *sysname* is the name of the MVS system where the ID runs the MODDVIPA utility or issues the SIOCSVIPA/SIOCSVIPA6 IOCTL.
   • *tcpname* is the job name of the TCP/IP started task.
   • *USER1* is the DDF user ID that is being permitted to run the MODDVIPA utility (and thus the SIOCSVIPA/SIOCSVIPA6 IOCTL).

If this security profile is created, the user ID must be permitted to access this profile. Otherwise, the SIOCSVIPA/SIOCSVIPA6 IOCTL (and thus the MODDVIPA utility) fails with a “permission denied” error, regardless of superuser authority.
Also note that when using RACF, you might need to refresh these profiles before they take effect. Use the following RACF command to refresh these profiles:

```
SETROPTS RACLST(SERVAUTH) REFRESH
```

7. Db2 for z/OS utilizes a TCP/IP DROP API through the EZBNMIFR callable service (NMI), and Db2 requires that RACF security profiles be defined to permit Db2 to successfully utilize this API. You must issue a RACF PERMIT command to enable ssnmDIST_userid CONTROL access to the MVS.VARY.TCPIP.DROP command for the OPERCMDS class, where ssnmDIST_userid is the user ID that is associated with the ssnmDIST started task.

Example (where the ssnmDIST_userid is SYS0):

```
RDEFINE OPERCMDS MVS.VARY.TCPIP.DROP OWNER(SYS0) UACC(NONE)
PERMIT MVS.VARY.TCPIP.DROP CLASS(OPERCMDS) ID(SYS0) ACCESS(CONTROL)
```

Failure to provide this permission might result in a ICH408I message similar to the following:

```
ICH408I USER(ssnmDIST_userid) ...
MVS.VARY.TCPIP.DROP CL(OPERCMDS) ...
```

For additional information, see [EZBNMIFR: Configuration and enablement](z/OS Communications Server: IP Programmer’s Guide and Reference).

**Related information:**


**Db2 configuration with TCP/IP**

The Db2 subsystem uses different TCP/IP ports to do different tasks.

- As a requester, Db2 uses an ephemeral port. You do not need to specify this port.
- As a server processing TCP/IP connection requests for DRDA SQL applications, Db2 uses a server port or the well-known port, 446, which is used for relational database communications.
- A server resynchronization port is used for processing 2-phase commit resynchronization requests.

This requires some planning because the port number is used to pass the network requests to the appropriate Db2 subsystem. Each location must have a unique port number.

The figure below shows some typical z/OS system configurations.

- SYSTEM1 contains only one Db2 subsystem, so the DRDA well-known port (446) can be assigned to Db2. In the example, port number 5020 is assigned for 2-phase commit resynchronization.
- SYSTEM2 contains two Db2 subsystems, so it is impossible to assign the port numbers 446 and 5020 to both Db2 subsystems because TCP/IP can only
support one server at each port number. The problem is resolved by assigning the 446 and 5020 port numbers to DB2C, and port numbers 5021 and 5022 to DB2D.

Be sure to consider the impact of future system consolidations. If SYSTEM1 and SYSTEM2 are consolidated so that DB2A, DB2C, and DB2D run on a single z/OS system, you must take special precautions because DB2A and DB2C have the same TCP/IP port numbers. You can resolve this conflict by changing the port numbers of either DB2A or DB2C to eliminate the duplicate port numbers.

Figure 138. Typical z/OS configurations

Related concepts:
- Reserving the DRDA port (Db2 Data Sharing Planning and Administration)

Customizing the TCP/IP data sets or files:

To use TCP/IP, you must customize your TCP/IP data sets or files.

Before you begin

UNIX System Services should already be installed. See the Db2 for z/OS Program Directories for required maintenance levels.

Procedure

To customize your TCP/IP data sets or files:
1. Find the high level qualifier (hlq) value in the resolve/tcpip.data file that z/OS UNIX applications use in their TCP/IP stacks. This data set defines the high level qualifier, which is added to the beginning of other data set names used by TCP/IP.
2. Find the `hlq.TCPPARMS(PROFILE)` data set. This data set contains the PORT statement used to make DRDA and resync port reservations. The following example shows a sample `hlq.TCPPARMS(PROFILE)` entry for SYSTEM2:

```
PORT 446 TCP DB2CDIST ; DRDA SQL port for DB2C
PORT 5020 TCP DB2CDIST ; Resync port for DB2C
PORT 5021 TCP DB2DDIST ; DRDA SQL port for DB2D
PORT 5022 TCP DB2DDIST ; Resync port for DB2D
```

This example assumes that `DB2CDIST` and `DB2DDIST` are the Db2 started procedure names.

3. Define the TCP/IP host names that Db2 needs to know. The local host name must be defined before DDF is started. All domain names referenced in the table SYSIBM.IP NAMES must be defined. You define the host names by configuring the `hlq.HOSTS.LOCAL` data set, the `/etc/hosts` file in the hierarchical file system (HFS), or the domain name server (DNS).

After the `hlq.HOSTS.LOCAL` data set is configured, you have to execute the utility MAKESITE. This utility generates the `hlq.HOSTS.ADDRINFO` and the `hlq.HOSTS.SITEINFO` data sets that are used to translate between domain names and IP addresses. MAKESITE generates under the userid that issued the MAKESITE. Therefore, those files have to be moved to the high level qualifier that represents TCP/IP. The host name for the local Db2 subsystem must be defined in at least one of these places.

You can also run the MAKESITE command with the HLQ parameter. You can set the HLQ parameter to the high level qualifier of the stack that you obtained in step 1. If you use the HLQ parameter, the `hlq.HOSTS.ADDRINFO` and the `hlq.HOSTS.SITEINFO` data sets are created and updated under the correct high level qualifier, so you do not have to move the data sets from under the userid high level qualifier to the TCP/IP high level qualifier.

If domain names are present in the CDB (in field IPADDR of table SYSIBM.IP NAMES), they must be defined in the z/OS data sets, the HFS or the DNS.

**Recommendation:** To support a Parallel Sysplex environment, use Dynamic Virtual IP Addresses (VIPA). To configure a Db2 subsystem to perform Dynamic VIPA routing, specify the Db2 group Dynamic VIPA on the TCP/IP PORT statement for the DRDA PORT number. This Dynamic VIPA must be the same for all Db2 members in the data sharing group. All clients must use the Dynamic VIPA to route requests to the Db2 group. To Db2 member-specific Dynamic VIPA is specified on the TCP/IP RESYNC PORT number in the TCP/IP profile data set for each Db2 member of the sysplex.

The following example shows a sample `hlq.TCPPARMS(PROFILE)` entry for SYSTEM2:

```
PORT 446 TCP DB2DIST BIND db2_sysplex_VIPA ; DRDA SQL port for DB2
PORT 5001 TCP DB2DIST BIND member_specific_VIPA ; RESYNC port for DB2
```

4. Define the TCP/IP service names that Db2 needs to know. Configure the `hlq.ETC.SERVICES` data set or the `/etc/services` file in the HFS. If service names are present in the CDB (in field PORT of table SYSIBM.LOCATIONS), they must be defined in the z/OS data set or the HFS.

The following example shows a sample `hlq.ETC.SERVICES` entry:

```
DRDA 446/tcp ; DRDA databases
```

Related concepts:

[Db2 configuration with TCP/IP]
Modifying the change log inventory job for using TCP/IP:

If you want to use TCP/IP, the DDF statement of the change log inventory job (DSNJU003) must specify values for the parameters PORT and RESPOR.

Procedure

To modify the change log inventory job for using TCP/IP:

Specify the PORT and RESPOR parameters in the DDF statement of DSNJU003. The parameter PORT is the TCP/IP port number that DDF uses to accept incoming DRDA connection requests. The parameter RESPOR is the TCP/IP port number that DDF uses to accept incoming DRDA 2-phase commit resynchronization requests. The values for these parameters must be decimal numbers between 0 and 65534, where 0 indicates that TCP/IP support is being deactivated. The non-zero value for PORT must not be the same as the non-zero value for RESPOR.

For data sharing, all members of the Db2 data sharing group must have the same value for PORT. RESPOR must be uniquely assigned to each Db2 member so that no two Db2 members use the same TCP/IP port for 2-phase commit resynchronization. The parameters PORT and RESPOR can be changed on any Db2 member by running the utility change log inventory. After running the utility, you must stop and then restart Db2. Because PORT is the same for all members of the Db2 group, this process has to be repeated on every member of the group when PORT is changed.

If you use Dynamic VIPA to support a Parallel Sysplex environment, specify the Db2 group Dynamic VIPA on the TCP/IP PORT statement for the DRDA PORT number.

You can define an alias location for all or selected members of a data sharing group by using the ALIAS parameter.

Remember, a zero value for either PORT or RESPOR is the same as deactivating TCP/IP support in Db2.

You can also specify an IPv4 address and an IPv6 address for the subsystem in the DDF statement of the change log inventory utility (DSNJU003). Specifying an IPv4 and an IPv6 address keeps the subsystem IP addresses constant, regardless of the z/OS system where the subsystem is started. The IP address moves with the Db2 subsystem. In data sharing environments, a constant IPv4 address and a constant IPv6 address can be assigned to represent the entire group.
Populating the communications database for use with TCP/IP
If you plan to use Db2 only as a server, you do not need to populate the CDB.

About this task

For example, Spiffy’s USIBMSTODb21 subsystem works as a server for many requesters. It is not necessary for Spiffy to register those requesters in the Db2 CDB.

However, if you intend to request data, you need to enter port numbers or service names in field PORT of table SYSIBM.LOCATIONS, and IP addresses or domain names in field IPADDR of table SYSIBM.IP NAMES. The LINKNAME in table SYSIBM.LOCATIONS is used to search tables SYSIBM.IP NAMES and SYSIBM.LUNAMES.

Updates to the following tables take place whenever a new remote connection is requested, without any need to stop and restart DDF:

- SYSIBM.LOCATIONS
- SYSIBM.IP NAMES
- SYSIBM.IPLIST

If you use RACF PassTickets, the LINKNAME must match the following values at the remote site:

- LUNAME
  If the remote site is a Db2 subsystem that is defined with only an LUNAME value and no GENERIC LU name value or IPNAME value.
- GENERIC
  If the remote site is a Db2 subsystem that is defined with a GENERIC LU name value in addition to an LUNAME value, but no IPNAME value.
- IPNAME
  If the remote site is a Db2 subsystem that is defined with an IPNAME value, which triggers the DDF on the remote Db2 subsystem to activate TCP/IP communications support only.

Procedure

- Insert rows into the SYSIBM.LOCATIONS table. The LOCATIONS table contains the port number or service name that is used to connect to remote locations. The LINKNAME column of the LOCATIONS table corresponds to a row in the IPNAMES table.
  The LOCATIONS table has the following columns that relate to TCP/IP:
  **DBALIAS VARCHAR(128) NOT NULL**
  The name that is associated with the server. This name is used to access a remote database server. If DBALIAS is blank, the location name is used to access the remote database server. This column does not change database object names that are sent to the remote by using a location qualifier. Use the DBALIAS column to access data at two or more different remote locations when those remote locations have the same name.
LOCATION CHAR(16)
The unique network location name, or DRDA RDBNAM, that is assigned to a remote or local system. You must provide location names for any systems from which you request data. This column is the primary key for this table.

LINKNAME CHAR(8)
Identifies the TCP/IP attributes that are associated with this location. For each specified LINKNAME, you must have a row in SYSIBM.IPNAMES whose LINKNAME matches the value that is specified in this column. Because this table is used for outbound requests, you must provide a LINKNAME or your requests fail. Do not enter blanks in this column.

PORT CHAR(32)
If blank, the default port, 446, is used for TCP/IP communications. Otherwise, the value can be either of the following values:
- The port number of the remote database server. The number must be one to five characters and left-justified.
- A TCP/IP service name. The service name is converted to a TCP/IP port number with the getservbyname socket call.

For example, Spiffy’s USIBMSTODB21 location expects a LOCATIONS table that looks like the table below. The location USIBMSTODB21 uses the default DRDA PORT, 446.

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>LINKNAME</th>
<th>PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>USIBMSTODB21</td>
<td>LUDB21</td>
<td></td>
</tr>
<tr>
<td>USIBMSTODB22</td>
<td>LUDB22</td>
<td></td>
</tr>
<tr>
<td>USIBMSTOSQL1</td>
<td>LUSQLDS</td>
<td>1234</td>
</tr>
<tr>
<td>USIBMSTOSQL2</td>
<td>LUSQLDS</td>
<td>DRDA</td>
</tr>
</tbody>
</table>

For example, add the second row with this statement:

```sql
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME) VALUES ('USIBMSTODB22', 'LUDB22');
```

Because no port number is specified, location USIBMSTODB22 uses the default DRDA port number 446.

Tip: You do not need a row for the local Db2 in the IPNAMES and LOCATIONS tables. For example, Spiffy’s USIBMSTODB21 subsystem does not require a row that shows its own LINKNAME and location name.

- Insert rows into the SYSIBM.IPLIST table. The IPLIST table contains a list of multiple IP addresses that are specified for a given location.
- IPLIST has the following columns:

  **LINKNAME CHAR(8) NOT NULL**
  This column is associated with the value of the LINKNAME column in SYSIBM.LOCATIONS and SYSIBM.IPNAMES. The values of the other columns in the SYSIBM.IPNAMES row apply to the server that is identified by the LINKNAME column in this row.

  **IPADDR VARCHAR(256) NOT NULL**
  This column contains an IPv4 or IPv6 address or domain name of a remote TCP/IP host of the server. If using WLM domain name server workload balancing, this column must contain the member-specific
domain name. If you use dynamic VIPA workload balancing, this column must contain the member-specific dynamic VIPA address.

- An IPv4 address must be left justified and is represented as a dotted-decimal address. An example of an IPv4 address is 9.112.46.111.

- An IPv6 address must be left justified and is represented as a colon-hexadecimal address. An example of an IPv6 address is 2001:0DB8:0000:0000:0008:0800:200C:417A, which can also be expressed in compressed form as 2001:DB8::8:800:200C:417A.

- A domain name is converted to an IP address by the domain name server where a resulting IPv4 or IPv6 address is determined. An example of a domain name is stlmvs1.svl.ibm.com. The gethostbyname socket call is used to resolve the domain name.

**IBMREQD CHAR(1) NOT NULL WITH DEFAULT 'N'**

This column indicates whether the row came from the basic machine-readable material (MRM) tape: N=no, Y=yes

- Insert rows into the SYSIBM.IPNAMES table. The IPNAMES table defines the outbound security and host names that are used to connect to other systems through TCP/IP.

  IPNAMES has the following columns:

**LINKNAME CHAR(8)**

This value matches that specified in the LINKNAME column of the associated row in SYSIBM.LOCATIONS.

**SECURITY_OUT CHAR(1)**

Defines the security option that is used when local Db2 SQL applications connect to any remote server associated with this TCP/IP host. The default, A, means that outgoing connection requests contain an authorization ID without a password.

**USERNAMES CHAR(1)**

This column is used for outbound requests to control translations of authorization IDs. The values 'O' or 'B' are valid for TCP/IP connections.

**IPADDR VARCHAR(254)**

This column contains an IPv4 or IPv6 address or domain name of a remote TCP/IP host.

- An IPv4 address must be left justified and is represented as a dotted-decimal address. An example of an IPv4 address is 9.112.46.111.

- An IPv6 address must be left justified and is represented as a colon-hexadecimal address. An example of an IPv6 address is 2001:0DB8:0000:0000:0008:0800:200C:417A, which can also be expressed in compressed form as 2001:DB8::8:800:200C:417A.

- A domain name is converted to an IP address by the domain name server where a resulting IPv4 or IPv6 address is determined. An example of a domain name is stlmvs1.svl.ibm.com. The gethostbyname socket call is used to resolve the domain name.

- Insert rows into the SYSIBM.USERNAMES table. The USERNAMES table contains information that is needed for outbound translation only.

**Remember:** Inbound ID translation and come from checking are not done for TCP/IP requesters.

The USERNAMES table has the following columns:
TYPE CHAR(1)
Whether the row is for outbound translation. The value ‘O’ is valid for TCP/IP connections.

AUTHID CHAR(8)
Authorization ID to translate. If blank, it applies to all authorization IDs.

LINKNAME CHAR(8)
Identifies the TCP/IP network location associated with the row. A blank indicates it applies to all TCP/IP partners. For nonblank values, this value must match the LINKNAME value in SYSIBM.IPNAMES.

NEAUTHID CHAR(8)
The translated value of AUTHID.

PASSWORD CHAR(8)
The password to accompany an outbound request. This column is ignored if RACF PassTickets, or already verified USERIDs are used.

What to do next
After you populate these tables, you can write queries that access data at a remote system.

Related concepts:
- Support for executing long SQL statements in a distributed environment (Db2 Application programming and SQL)

Related tasks:
- Granting privileges to remote users (Managing Security)

Related reference:
- SYSIBM.USERNAMES columns (Managing Security)

Starting TCP/IP support
Before you start DDF, ensure that z/OS UNIX System Services and TCP/IP are started. Also ensure that the local host information is set up correctly in the TCP/IP configuration files.

About this task
Important: When using dynamic VIPA, the IP addresses that have been assigned to Db2 by the TCP/IP profile must be registered with the DNS.

Procedure
To start TCP/IP support:

Start DDF.
DDF executes the following steps when it is started:
1. Establishes the TCP/IP listener by opening sockets for the DRDA SQL, Secure SQL, and resync ports to accept connections from remote locations.
2. Obtains the host name that is specified in TCPIP.DATA. The host name is used to register with WLM and, in some cases, to obtain the IP address. You can specify the IP address that Db2 should use for communication with its partners as follows:
• Note that Db2 12 is an IPv6 system and does not use gethostid to obtain the IP addresses because gethostid does not support IPv6. You can instead specify the address in the domain name server (DNS), if one is present, or in a local host file. You can specify an IPv4 address, an IPv6 address, or both, which should be mapped to the host name that is specified in TCP/IP DATA. Db2 obtains the user-specified IP addresses by resolving the host name.

• If you want DDF to accept connections only on a specific IP address (IPv4 or IPv6), specify the IP address on the SQL port in the TCP/IP profile by using the BIND keyword. For example, use the following configuration, where 446 is the SQL port and Vx is either an IPv4 or IPv6 address:

   PORT
   446 TCP DB2ADIST BIND Vx

With this method, Db2 only accepts connections on the specified IP address. Moreover, this method is not supported with secure SQL port. If your subsystem is running in data sharing mode, you must also specify an IP address on the DRDA resync port. For more details, see TCP/IP access methods (Db2 Data Sharing Planning and Administration).

• The most recommended place to specify the IP address is in the DDF communication record of the bootstrap data set (BSDS) by using the DSNJU003 utility. For example:

   //SAMPLE EXEC PGM=DSNJU003
   //SYSIN DD *
   DDF IPV4=1.2.3.4,
   IPV6=1::2

The advantage of this approach is that Db2 accepts connections not only on the specified IP address, but on any IP address that is active on the TCP/IP stack. Additionally, connections are accepted on both secure and non-secure SQL ports. Db2 also resolves the host name to an IP address through a name server or local host file to fill up any IP address that is not specified in the BSDS. For example, if the BSDS specified IPv4 but not IPv6, Db2 would try to obtain the IPv6 address by resolving the host name.

Important: IP addresses should not be specified simultaneously in both the Db2 BSDS and TCP/IP PORT statement. Binding to a specific IP address and binding to any IP address (INADDR_ANY) are mutually exclusive, and DDF TCP/IP services will not be available until the IP addresses are specified in only one place.

3. Obtains the domain name by resolving the IPv6 address, if one is specified, or the IPv4 address through a name server or the local host file. If you use data sharing, member domain names are obtained by resolving member IP addresses, and group domain names are obtained by resolving group IP addresses.

Important: Whether you use data sharing or not, Db2 12 uses the domain name that you specify in the name server, if one is present, or in a local host file, without further modifying or qualifying the name. However, DDF tolerates the absence of a domain name, when the IP address is specified in the Db2 BSDS.

To determine the host IP address and domain name, issue a DISPLAY DDF command. The IPADDR field of the DSNL085I message displays the IPv4 and IPv6 addresses, if any, that are associated with the host.
Until the local host information is available, DDF TCP/IP services are not available to local and remote applications. If a failure occurs obtaining the local host information, DDF periodically attempts to get the local host information until successful or until DDF is stopped. DDF tolerates the absence of a domain name as long as an IP address is specified in the Db2 BSDS.

4. Listens and accepts connection requests on SQL, secure SQL, and resync ports.

Related concepts:

- [Communicating with data sharing groups (Db2 Data Sharing Planning and Administration)](#)
- [DVIPA network addressing (Db2 Data Sharing Planning and Administration)](#)

**Tuning TCP/IP**

Tuning TCP/IP is optional, but it can protect Db2 from TCP/IP outages.

**Procedure**

To tune TCP/IP:

Specify a small value, five minutes or less, for the TCP/IP KeepAlive timer. You can specify the value for the TCP/IP KeepAlive timer either in the TCP/IP KEEPALIVE field of the Distributed data facility panel 2: DSNTIP5 or in the TCPKPALV parameter. If the network fails between the server's reply and the next client request, TCP/IP waits until the KeepAlive timer expires, then notifies the Db2 subsystem of the failure.

The server thread hangs while the timer is running. Do not use the timer default of ENABLE because it results in a value of 2 hours, which means that threads can hang for up to 2 hours. A hanging thread can cause unpredictable results, depending on what resources it has locked.

If you are connecting to a location whose LINKNAME is associated with a row in the table SYSIBM.IPNAMES and it has a domain name in the IPADDR field, gethostbyname can return a list of IP addresses associated with the LINKNAME. When trying to connect to this location, Db2 will try each of these IP addresses in a round-robin fashion (starting with the first address) until the connection is successful, or the attempt to connect to each IP address has timed out.

**TCP/IP limitations**

TCP/IP does not have the built-in security features that SNA has, such as SNA partner LU verification.

Because IP addresses are not as reliable as LU names, DDF support for TCP/IP differs from support for SNA in these ways:

- There is no support for inbound name translation.
- A Db2 subsystem parameter defines the minimum security requirements for all TCP/IP clients because inbound security requirements cannot be established on individual clients. See the description of the TCP/IP ALREADY VERIFIED field on installation panel DSNTIP5.
You cannot use the CDB for come from checking of TCP/IP clients.

Related reference:
[DSNTIP5: Distributed data facility panel 2]

Initializing a TCP stack for use with a VIPA

During initialization of a TCP stack for use with a virtual IP address (VIPA), the PROFILE.TCPIP configuration should contain the HOME list with the VIPA. The HOME list should be followed by the PRIMARYINTERFACE statement pointing at the appropriate VIPA that is to be used by Db2.

About this task

Recommendation: Do not change the stack’s HOME list or the PRIMARYINTERFACE while DDF is started.

Procedure

To initialize a TCP stack for use with a VIPA:

Use the VARY OBEY command to change the PRIMARYINTERFACE and the stack’s HOME list. This command can be used while the stack is running without recycling the stack. If a new HOME list is specified with the VARY OBEY file, the PRIMARYINTERFACE should also be specified.

Initializing a TCP stack for use with both IPv4 and IPv6 addresses

To use both IPv4 and IPv6 addresses, Db2 requires TCP/IP dual-mode stack support. Dual-mode stack support allows IPv4 address communication with IPv4 partners, and IPv6 address communication with IPv6 partners.

About this task

If you need only IPv4 address communication, an IPv4-only stack environment is sufficient because Db2 detects if the stack supports IPv6, and if not, Db2 allows communication with IPv4 addresses only.

Procedure

To enable TCP/IP for IPv4/IPv6 dual-mode stack:

1. Modify the BPXPRMxx member to define two NETWORK statements, one for AF_INET, and another for AF_INET6.

   The following example shows two NETWORK statements in the BPXPRMxx member:

   ```
   TYPE(INET) ENTRYPOINT(EZBPFINI)
   NETWORK DOMAINNAME(AF_INET)
     DOMAINNUMBER(2)
     MAXSOCKETS(12000)
     TYPE(INET)
   NETWORK DOMAINNAME(AF_INET6)
     DOMAINNUMBER(19)
     MAXSOCKETS(13000)
     TYPE(INET)
   SUBFILESYSTYPE NAME(TCPIP) ENTRYPOINT(EZBPFINI)
   TYPE(INET)
   ```
Separate MAXSOCKETS values are supported. The IPv6 default will be the IPv4 specified value.

2. To verify that TCP/IP is configured to support a dual-mode environment, enter a `d tcpip,,netstat,home` command on the z/OS console. Look for the INTFNAME field that has a value of LOOPBACK6 and verify that the associated ADDRESS field has a compressed colon-hexadecimal IPv6 address of ::1.

The following example shows that TCP/IP is configured to support both IPv4 and IPv6:

d tcpip,,netstat,home
EZD010II NETSTAT CS V1R5 TCPIP 034
HOME ADDRESS LIST:
  LINKNAME: CTC1LINK
    ADDRESS: 9.30.115.135
    FLAGS: PRIMARY
  LINKNAME: LOOPBACK
    ADDRESS: 127.0.0.1
    FLAGS: 
  INTFNAME: LOOPBACK6
    ADDRESS: ::1
    TYPE: LOOPBACK
    FLAGS: 3 OF 3 RECORDS DISPLAYED

Related reference:
[2/OS Communications Server: IPv6 Network and Application Design Guide](#)

Two-phase commit for TCP/IP clients

Db2 supports two types of two-phase commit for TCP/IP clients.

- The DRDA client coordinates the two-phase commit. If a failure occurs during the commit process, Db2 might need to resynchronize with the DRDA client.
- The DRDA client gives responsibility for the resynchronization to Db2. The client sends Db2 a list of server LOCATION names, domain names, and resync IP addresses that are part of the client's unit of work. Db2 Connect uses this support by allowing Db2 to act as its Transaction Manager Database (TM_DATABASE), thereby eliminating the need to have a local database to manage the two-phase commit process. If a failure occurs during the two-phase commit process, Db2 might need to resynchronize with one or more of the server locations sent by the client.

Db2 uses the port specified on the RESYNC PORT field of installation panel DSNTIP5 for two-phase commit resynchronization. Db2 begins resynchronization using the partner's IP address and LOCATION name, and the RESYNC PORT obtained at the time of initial connection. If the partner's IP address changed, the resynchronization fails. For example, if the partner was Db2 for z/OS, the IP address can change when the automatic restart manager (ARM) restarts a data sharing member on a different CPC. The IP address can also change when an z/OS adapter fails and virtual IP addresses were not used.

If the IP address fails, Db2 uses the partner's domain name to determine the IP address for resynchronization. DRDA requesters receive the port number and domain name to be used for two-phase commit resynchronization from the server during DRDA connect processing.

No CDB definition is required to do resynchronization.
Multiple TCP/IP stacks and Db2

Running multiple TCP/IP stacks does not provide performance advantages over a single TCP/IP stack, which can provide high levels of scalability and performance by itself. Running multiple stacks requires more private and ECSA memory consumption and some additional processor consumption. It is also a much more complicated environment to manage from an administrative perspective.

Although multiple TCP/IP stacks are not recommended for Db2, multiple stacks might be appropriate for your installation. A possible scenario for using multiple TCP/IP stacks could be where a single Db2 system must connect to multiple networks that have distinct security characteristics (for example, intranet and internet) and strict separation is required. The firewall on the external stack can be configured to forward only selected traffic internally, while the internal stack does what is indicated with internal traffic, including forwarding to the external stack. Second, a denial-of-service attack on the stack that handles external traffic would not affect the stack that handles internal traffic, so two stacks offer an element of resistance to attack.

Multiple Db2 subsystems with multiple TCP/IP stacks

z/OS UNIX System Services provides a way to configure Db2 to use a single transport stack.

This allows each Db2 subsystem to be restricted to a single stack. Each would be reachable by any IP address that is owned by the selected stack, but not by other IP addresses. Adding BPXTCAFF as a new job step before the existing EXEC card for DSNYASCP for the ssnmDIST procedure allows Db2 to bind to a single stack when there are at least two stacks running on a single z/OS image. When you add the BPXTCAFF step as the first step, you must specify either TIME=1440 or TIME=NOLIMIT to the EXEC DSNYASCP card because the SYST parameter is disabled when a second job step is added to the ssnmDIST procedure.

Multiple Db2 subsystems with one TCP/IP stack

The TCP Server Bind Control allows multiple Db2 subsystems to use the same port number within the same z/OS image.

This allows multiple Db2 subsystems that bind to any IP address (also known as INADDR_ANY in sockets API terms) and the same port number on the same stack to be bound to separate IP addresses. For example, consider two Db2 subsystems in a data sharing group where both have the same LOCATION and same DRDA PORT. A new 'BIND ipaddr' parameter is added to the PORT statement. When Db2 (identified by job name ssnmDIST) issues a bind to the port number in the PORT statement and to INADDR_ANY, the bind is restricted to the IP address specified on the PORT statement for that Db2 subsystem. This allows two Db2 subsystems to share the same stack by limiting them to different single IP addresses (virtual IP addresses are recommended). This allows them to be reached by any physical connectivity that can get to the stack.

Related reference:

PORT statement (z/OS Communications Server: IP Configuration Reference)

Specifying security requirements

When you set up systems to communicate through TCP/IP, you must specify security requirements.

Related tasks:
Setting up the Db2 command line processor

You can use the Db2 command line processor to issue SQL statements, bind DBRMs that are stored in HFS files, and call stored procedures. The Db2 command line processor on Db2 for z/OS is a Java application that runs under z/OS UNIX System Services.

Before you begin

Before you begin setting up the Db2 command line processor, ensure that your system has the following features:

- Java 1.4.2 service release 2. If you use or reference the Db2 12 DECIMAL FLOAT SQL type, Java 1.5 is required.
- IBM Data Server Driver for JDBC and SQLJ.
  For information on installing this feature, see [Installing the IBM Data Server Driver for JDBC and SQLJ on Db2 for z/OS (Db2 Application Programming for Java)](function).
- Db2 metadata stored procedures installed and authorized. To install and authorize the Db2 metadata stored procedures, run the installation job DSNTIJRT. To check if the Db2 metadata stored procedures were successfully installed and activated, run the validation job DSNTIJRV.

Procedure

To set up the Db2 command line processor, complete the following steps:

1. Modify the CLASSPATH variable in your .profile file to include the clp.jar file. If you install the Db2 command line processor into the /usr/lpp/db2c10_base directory, you can issue the following command to modify your CLASSPATH variable:

   ```
   export CLASSPATH=$CLASSPATH:/usr/lpp/db2c10_base/lib/clp.jar
   ```

2. Optional: Define the CLPPROPERTIESFILE environment variable in your .profile file to include the full path name where the properties file is located. You can create your own properties file with any name. A sample properties file is in the directory where the command line processor is installed. If you install the command line processor into the /usr/lpp/db2c10_base directory, you can issue the following command to define the CLPPROPERTIESFILE environment variable in your .profile file:

   ```
   export CLPPROPERTIESFILE=/usr/lpp/db2c10_base/samples/clp.properties
   ```

   If the CLPPROPERTIESFILE environment variable is undefined, the command line processor uses default values for all options when it is invoked.

3. Optional: Define an alias for starting the Db2 command line processor in your .profile file: For example:

   ```
   alias db2="java com.ibm.db2.clp.db2"
   ```

Results

Additional instructions for setting up the Db2 command line processor are available in the readme file. By default, the Db2 command line processor readme file (README_CLP) is installed into the /usr/lpp/db2c10_base directory.

Related concepts:
IBM Text Search for Db2 for z/OS

IBM Text Search for Db2 for z/OS provides full-text search capability for text columns that are stored in Db2.

Creating database objects that support query acceleration

If you use an accelerator, you must create a database, table spaces, tables, and indexes for the accelerator to use.

Before you begin

1. Apply or verify that you have applied the PTF for APAR PI91467 (UI54612) and for APAR PI19468 (UI54615). For a data sharing group, these PTFs must be applied on all members of the group.

2. After the PTFs for PI91467 have been applied, restart Db2 or, for a data sharing group, restart each member of the group.

Procedure

To create database objects for use by an accelerator:
1. Configure and run the DSNTIJAS sample job in the SDSNSAMP library to create the objects that are required for an accelerator.

After you run DSNTIJAS:

- The SYSACCEL.SYSACCELERATORS and SYSACCEL.SYSACCELERATEDTABLES tables are created. The accelerator will populate these tables with a server definition and the associated IP address.

- The SYSACCEL.SYSACCELERATEDPACKAGES table and its indexes are created. This table and its indexes will be populated by Db2 during a bind operation that specifies the QUERYACCELERATION bind option. They are not used by the accelerator.

- The SYSACCEL.SYSACCELERATEDTABLESAUTH table and its index are created for federation support. This table and its index will be populated by the IBM Db2 Analytics Accelerator for z/OS SYSPROC.ACCEL_GRANT_TABLES_REFERENCE and SYSPROC.ACCEL_REVOKE_TABLES_REFERENCE stored procedures.

2. Restrict access to the SYSACCEL.SYSACCELERATORS, SYSACCEL.SYSACCELERATEDTABLES, and SYSACCEL.SYSACCELERATEDPACKAGES tables that were created by job DSNTIJAS.

The SYSACCEL.SYSACCELERATORS and SYSACCEL.SYSACCELERATEDTABLES tables are used by Db2 to control the acceleration behavior of the accelerator servers that are defined for the Db2 subsystem.
The SYSACCELERATEDPACKAGES table is used by Db2 to control the acceleration behavior of static SQL queries in a Db2 package.

Related tasks:

- [Enabling Db2 for IBM Db2 Analytics Accelerator for z/OS (Db2 Performance)]

Related reference:

- [Support Portal: IBM Db2 Analytics Accelerator for z/OS]
- [Tables that support query acceleration (Db2 SQL)]

---

## Migrating query acceleration tables to universal table spaces

This topic provides instructions for migrating query acceleration tables from a single segmented table space (SYSACCEL) to individual universal table spaces (UTSs).

### Before you begin

**Attention:** Before you migrate the acceleration tables to universal tables spaces, make sure that an accelerator has already been defined on your system. If you do not already have an accelerator defined on your system, defer this migration task until an accelerator has been defined.

### About this task

Migrating the query acceleration tables to individual UTSs is optional and can be deferred. However, if you are experiencing frequent timeout issues on the query acceleration tables, you should consider migrating.

If you choose to migrate the query acceleration tables from a single segmented table space to individual UTSs, you must complete all three of the following subtasks:

1. Complete the pre-migration steps, which are documented in this topic.
2. Run job DSNTIJAM, which migrates the tables.
3. Complete the post-migration steps, which are also documented in this topic.

### Procedure

Complete the following pre-migration steps:

1. Apply the PTF for APAR PI96858 (UI56078) and for APAR PI96859 (UI56897), or verify that these PTFs are applied. In a Db2 data sharing group, these PTFs must be applied on all members of the group.
2. After the PTFs for APAR PI96858 are applied, restart Db2 or, for a data sharing group, restart each member.
3. Verify that the following Db2 query acceleration tables are present in the DSNACCEL.SYSACCEL table space and that they are at same level as PI91620 (UI54293):
   - SYSACCEL.SYSACCELERATORS
   - SYSACCEL.SYSACCELERATEDTABLES
   - SYSACCEL.SYSACCELERATEDPACKAGES
   - SYSACCEL.SYSACCELERATEDTABLESAUTH
   Also, verify that column 13 of the SYSACCEL.SYSACCELERATEDTABLES table is the REMOTELOCATION column.
4. Drop, re-create, and rename any user-defined table spaces in the DSNACCEL database that are named SYSTSACC, SYSTSACT, SYSTSACP, or SYSTSA. These names are reserved for the new UTSs in which the Db2 query acceleration tables will reside.

5. Drop any user-defined objects in table space DSNACCEL.SYSACCEL, and re-create them in another table space. When you run the DSNTIJAM job to migrate the query acceleration tables, the DSNACCEL.SYSACCEL table space and all of its objects will be dropped and re-created. However, only the following query acceleration tables and indexes will be re-created in the appropriate table space:
   - SYSACCEL.SYSACCELERATORS
   - SYSACCEL.SYSACCELERATEDTABLES
   - SYSACCEL.SYSACCELERATEDPACKAGES
   - SYSACCEL.SYSACCELERATEDTABLESAUTH
   - SYSACCEL.DSNACC01
   - SYSACCEL.DSNACT01
   - SYSACCEL.DSNACP01
   - SYSACCEL.DSNACP02
   - SYSACCEL.DSNATA01

   Therefore, if you created any user-defined objects in the DSNACCEL.SYSACCEL table space other than the ones in the previous list, you must drop and re-create them in another table space.

6. Rename any table that has the same name as the following tables. These names are reserved for the intermediate query acceleration tables that will be used during migration.
   - SYSACCEL.SYSACCELERATOR1
   - SYSACCEL.SYSACCELERATEDTABLE1
   - SYSACCEL.SYSACCELERATEDPACKAGE1
   - SYSACCEL.SYSACCELERATEDTABLESAUTH1

7. Rename any index that has the same name as the following indexes. These names are reserved for the indexes of the intermediate query acceleration tables that will be used during migration.
   - SYSACCEL.DSNACC1
   - SYSACCEL.DSNACT1
   - SYSACCEL.DSNACP1
   - SYSACCEL.DSNACP2
   - SYSACCEL.DSNATA1

8. Save any dependencies and authorization records for the following Db2 query acceleration objects so that you can re-create them after migration. These dependencies can include views, triggers, and masks, as well as authorizations, referential integrity definitions, check constraints, history versioning, and comments.
   - SYSACCEL.SYSACCELERATORS
   - SYSACCEL.DSNACC01
   - SYSACCEL.SYSACCELERATEDTABLES
   - SYSACCEL.DSNACT01
   - SYSACCEL.SYSACCELERATEDPACKAGES
   - SYSACCEL.DSNACP01
Additionally, if the DSN6SPRM.ABIND subsystem parameter is set to NO, identify all packages that are dependent on the DSNACCEL.SYSACCEL table space. You must explicitly rebind these packages after you complete the migration.

**Attention:** Because the DSNACCEL.SYSACCEL table space will be dropped during migration, failure to complete this step prior to running DSNTIJAM can result in the loss of administrative control of the accelerator and render it unusable and not easily recoverable.

9. Set the Db2 DSN6SPRM.QUERY_ACCEL_OPTIONS subsystem parameter to a value of 98, which is a special serviceability option that indicates that the migration of the query acceleration tables is in progress. For a data sharing group, set this subsystem parameter to 98 for each member of the data sharing group. Setting this subsystem parameter to 98 eliminates interference from other Db2 workloads.
   a. Create a temporary copy of your current customized copy of the DSNTIJUZ job.
   b. Edit the temporary copy of DSNTIJUZ and set the QUERY_ACCEL_OPTIONS parameter to a value of 98. If necessary, remember to add a continuation character in column 72.
   c. Run the first two steps of the temporary DSNTIJUZ job.
   d. After the job completes, issue the -SET SYSPARM command, or stop and restart Db2 to enable this change.
   e. Delete the temporary copy of DSNTIJUZ.

10. Issue the STOP ACCEL command (all query offload), stop all query accelerator data maintenance (incremental update and stored procedure jobs), and disconnect Data Studio from the database. For a data sharing group, complete these steps for each member of the data sharing group.

**What to do next**

After you successfully complete the pre-migration steps, customize and run the DSNTIJAM job to migrate the table spaces, and confirm that it completes with a return code of 0. Then, return to this topic and complete the following post-migration steps to re-enable query acceleration.

1. Run the AQTIIJSP job with the MODE(INSTALL) parameter.
2. Reset the DSN6SPRM.QUERY_ACCEL_OPTIONS subsystem parameter back to its original value to indicate that the migration of the query acceleration tables is no longer in progress:
   a. Run the first two steps of your original customized copy of job DSNTIJUZ.
   b. After the job completes, either issue the -SET SYSPARM command or stop and restart Db2 to enable this change.
3. Re-create the dependencies and authorization records that you saved in step 8 on page 938 of the pre-migration steps.
   If the DSN6SPRM.ABIND subsystem parameter is set to NO, explicitly rebind the packages that you identified in step 8 on page 938.
4. Issue the START ACCEL command, start all query accelerator data maintenance (incremental update and stored procedure jobs), and reconnect to Data Studio. For a data sharing group, complete these steps for each member of the data sharing group.

Related tasks:

- [Enabling Db2 for IBM Db2 Analytics Accelerator for z/OS (Db2 Performance)]

Related reference:

- [Support Portal: IBM Db2 Analytics Accelerator for z/OS]
- [Tables that support query acceleration (Db2 SQL)]
Chapter 12. Working with international data

Db2 handles different character sets in distributed data through character conversions.

Character conversion concepts

Character data that is transmitted from one DBMS to another might need to be converted to a different coded character set.

In different database management systems (DBMSs), character data can be represented by different encoding schemes. Within an encoding scheme, there are multiple coded character set identifiers (CCSIDs). EBCDIC, ASCII, and Unicode are ways of encoding character data.

The Unicode character encoding standard is a character encoding scheme that includes characters from almost all living languages of the world. Db2 supports two implementations of the Unicode encoding scheme: UTF-8 (a mixed-byte form) and UTF-16 (a double-byte form).

All character data has a CCSID. Character conversion is described in terms of CCSIDs of the source and of the target. When you install Db2, you must specify a CCSID for Db2 character data in either of the following situations:

- You specify AUTO or COMMAND for the DDF STARTUP OPTION field on panel DSNTIPR.
- Your system will have any ASCII data, Unicode data, EBCDIC mixed character data, or EBCDIC graphic data. In this case, you must specify YES in the MIXED DATA field of panel DSNTIPF, and the CCSID that you specify is the mixed data CCSID for the encoding scheme.

The CCSID that you specify depends on the national language that you use.

Db2 performs most character conversion automatically, based on system CCSIDs, when data is sent to Db2 or when data is stored in Db2. If character conversion must occur, Db2 uses the following methods:

1. Db2 searches the catalog table SYSIBM.SYSSTRINGS.
2. Db2 uses z/OS Unicode Conversion Services.

If Db2 or z/OS Unicode Conversion Services does not provide a conversion for a certain combination of source and target CCSIDs, you receive an error message. If the conversion is incorrect, you might get an error message or unexpected output. To correct the problem, you need to understand the rules for assigning source and target CCSIDs in SQL operations.

Related concepts:

- System-coded character set identifiers
- Unicode support in Db2
- Coded character sets and CCSIDS (Introduction to Db2 for z/OS)

Related reference:

- Character conversion (z/OS: Unicode Services User’s Guide and Reference)
System-coded character set identifiers

To support character conversion, the IBM Distributed Relational Database Architecture (DRDA) uses CCSIDs to label the various character representation schemes.

The CCSID is a two-byte binary number that uniquely identifies one or more pairs of character sets and code pages. The coded character set defines how bit configurations are mapped for character data.

The CCSID of character strings at your site is determined by the CCSID that you specify on installation panel DSNTIPF.

The correct CCSID is the number that identifies the coded character set that is supported by your I/O devices, local applications such as IMS and QMF, and remote applications such as CICS Transaction Server.

Attention: If this CCSID is not correct, character conversion produces incorrect results.

Related reference:

DSNTIPF: Application programming defaults panel 1

Related information:

Character Data Representation Architecture Reference

Unicode support in Db2

Unicode is a universal encoding scheme for written characters and text that enables the exchange of data internationally. Unicode provides a character set standard that can be used all over the world.

Unicode uses an encoding scheme that provides code points for approximately 100,000 characters. An extension called UTF-16 allows for encoding as many as a million more characters. Unicode provides the ability to encode all characters used for the written languages of the world. Unicode treats alphabetic characters, ideographic characters, and symbols equivalently because it specifies a numeric value and a name for each of its characters. Unicode includes punctuation marks, mathematical symbols, technical symbols, geometric shapes, and dingbats.

Db2 provides the following Unicode encoding forms:

- UTF-8: Unicode Transformation Format, 8-bit encoding form that is designed for ease of use with existing ASCII-based systems.
- UTF-16: Unicode Transformation Format, 16-bit encoding form that is designed to provide code values for over a million characters and a superset of UCS-2. UCS-2 is a universal character set and is coded in 2 octets, which means that characters are represented in 16 bits per character.

Unicode CCSIDs: The Unicode CCSID field of panel DSNTIPF is pre-filled with 1208. Db2 chooses the CCSIDs for double-byte and single-byte values (1200 for DBCS and 367 for SBCS). CCSID 1200 corresponds to UTF-16 and CCSID 367 is for 7-bit ASCII.

Customizing support for Unicode

You must customize z/OS Unicode support to use Unicode data in Db2.
Before you begin

Ensure that the conversion environment is active. The steps for this process can be found in 

**z/OS Support for Unicode: Using Unicode Services**

Db2 can use the conversion services of z/OS support for Unicode only when the conversion environment is active. The infrastructure provides tools to create a conversion image. When the image is loaded into a common data space, the conversion environment is activated, and the conversion services are ready to be used by Db2.

Procedure

To customize support for Unicode:

1. Customize the job card. The jobs in hlq.SCUNJCL, as shipped by IBM, have placeholders for values on the JOB statement, such as in the following example:

```bash
//JOBPREFS$JOBNAME$ JOB ($ACCOUNT$), '$USER$',
// NOTIFY=$NOTIFY$,MSGCLASS=$MC$,MSGLEVEL=$ML$,
// TIME=$TI$,CLASS=$CL$,REGION=$REGION0M$
```

Use the REXX EXEC CUNRUCST in hlq.SCUNREXX to customize these values. When you run the REXX EXEC CUNRUALL, the values that you specify are supplied on all of the JCL images. You can choose to customize your system by displaying Japanese messages or displaying English messages with modified date and time formats. You can set up the z/OS Message Service to specify how you want messages to be displayed.

2. Set up the conversion image. The following example is the sample JCL member in hlq.SCUNJCL (CUNJIUTL):

```bash
//CUNMIUTL EXEC PGM=CUNMIUTL
//SYSPRINT DD SYSPUT=* *
//TABIN DD DISP=SHR DSN=hlq.SCUNTBL
//SYSIMG DD DSN=hlq.IMAGES(CUNIMG00),DISP=SHR
//SYSIN DD *
/********************************************************************************
* INPUT STATEMENTS FOR THE IMAGE GENERATOR *
*********************************************************************************
CASE NORMAL; /* ENABLE TOUNPER AND TOLOWER */
CONVERSION 1047,850; /* EBCDIC -> ASCII */
CONVERSION 850,1047; /* ASCII -> EBCDIC */
/*
In the preceding example, the two CONVERSION statements provide conversion between the EBCDIC code page 1047 and the ASCII code page 850 in both directions. The DD names that are passed to CUNMIUTL are described as follows:

**SYSPRINT**
A listing that shows the processed setups and error messages, if applicable.

**TABIN**
Conversion tables for character conversion and case conversion. They are supplied by IBM; in this example, they are in data set hlq.SCUNTBL. The image transforms the conversion tables into an internal format and stores them in the conversion image.

**SYSIMG**
Output is a single image of the entire conversion environment. The conversion image is built according to the specification in the SYSIN DD name. The conversion image resides in either a sequential data set
or a member of a partitioned data set with a fixed-block 80-byte format. In this example, the image resides in the partitioned data set member hlq.IMAGES(CUNIMG00).

SYSIN

Two types of statements are recognized in this DD statement: case conversion, which is identified by the CASE control statement, and character conversion, which is identified by the CONVERSION control statement.

CASE control statement:
Case conversion is defined as converting Unicode characters (for example, UTF-8) to their uppercase equivalent or their lowercase equivalent. In the preceding example, CASE NORMAL is specified, which means basic case conversion is provided. This basic case conversion is based on the UnicodeData.txt file that is provided by the Unicode consortium. It does not include special casing as described in the SpecialCasing.txt file that is provided by the Unicode consortium. Special casing typically includes characters that have significant differences in the case-based appearance. For example, the German "hard S", which appears as a flat B, appears as "SS" in uppercase German text. Because Db2 does not use the case conversion service, you do not need to specify a conversion.

CONVERSION control statement:
Character conversion is also referred to as conversion between specified CCSIDs. An application such as Db2 invokes the CUNLCNV function to convert characters between the specified code pages. You must identify the conversions that are possible on the CONVERSION control statement.

Important: Specify CONVERSION statements for Db2 as follows:
CONVERSION xxx,yyy,ER;
CONVERSION yyy,xxx,ER;

Many code page conversions are possible. However, when identifying the conversion that Db2 is to use, you need be concerned only with conversion for the national languages that you use and with conversion from these code pages to and from all Unicode CCSIDs.

Example: If you use an EBCDIC CCSID of 37 and an ASCII CCSID of 819, you need to use the following conversions:
CONVERSION 37,367,ER;
CONVERSION 37,1208,ER;
CONVERSION 37,1200,ER;
CONVERSION 367,37,ER;
CONVERSION 1208,37,ER;
CONVERSION 1200,37,ER;
CONVERSION 819,367,ER;
CONVERSION 819,1208,ER;
CONVERSION 819,1200,ER;
CONVERSION 367,819,ER;
CONVERSION 1208,819,ER;
CONVERSION 1200,819,ER;
Multiple conversion tables might be available for converting one CCSID to another. A technique search order can be used to specify which table should be used. The technique search order consists of up to eight technique characters. If you specify more than one technique character, the image generator tries to find a matching table for the leftmost technique character in the sequence of the technique-search-order. If one is not found, the search continues with the second one, and so on. Especially for mixed conversion, use more than one technique character because one of the subconversions might exist only in round-trip mode, and one might exist only in an enforced subset. In this case, a technique search order of 'RE' or 'ER' would be required. Technique search order is optional. If you do not specify a technique search order, RECLM is used.

Language products such as Enterprise Cobol might use the RECLM technique search order, while Db2 uses the ER technique search order. Therefore, you might also need to add the RECLM conversions, such as these:

\[
\begin{align*}
\text{CONVERSION} & 1047,850,\text{RECLM}; \\
& /* \text{EBCDIC} \rightarrow \text{ASCII} */ \\
\text{CONVERSION} & 850,1047,\text{RECLM}; \\
& /* \text{ASCII} \rightarrow \text{EBCDIC} */ \\
\end{align*}
\]

The important technique characters for Db2 are E (enforced subset) and R (round-trip). Enforced subset conversions map only those characters from one CCSID to another that have a corresponding character in the second CCSID. All other characters are replaced by a substitution character. Round-trip conversions between two CCSIDs assure that all characters making the 'round trip' arrive as they were originally, even if the receiving CCSID does not support a given character. Round-trip conversions ensure that code points that are converted from CCSID A to CCSID B, and back to CCSID A are preserved, even if CCSID B is not capable of representing these code points.

After performing these steps, you should now have an updated CUNJIUTL JCL member.

3. Submit the batch job in the CUNJIUTL member. At completion, the batch job writes its output to the SYSPRINT DD (that is, SYSOUT in this example). Expect a return code of zero from the CUNJIUTL program. If you receive anything other than return code zero, refer to Return code meanings. This information helps you correct environmental, syntactical, and semantic errors that might occur.

4. After generating the conversion image, copy it to SYS1.PARMLIB or any other data set in the logical PARMLIB concatenation. In this example, you copy hlq.IMAGES(CUNIMG00) to SYS1.PARMLIB(CUNIMG00).

5. Calculate the storage that is needed for a conversion image. When the conversion image is created on disk, you need to determine the amount of virtual storage that the image is to occupy. You specify this number as the number of pages on the REALSTORAGE parameter in the CUNUNIx PARMLIB member that you create in the next step. The REALSTORAGE parameter protects the system from a shortage of main storage caused by loading a conversion image that exceeds the amount of available storage. The minimum value for the REALSTORAGE parameter depends on how the image is activated.
• If the image is activated during IPL, the needed storage is the size of the image plus one page.
• If the image is activated using the SET UNI command (PARMLIB member with keyword IMAGE), the needed storage is the size of the currently active image plus the size of the new conversion image.

If you set up a conversion environment before or if you are not activating a conversion environment during IPL, you must determine the amount of storage that the currently active image occupies. To do this, issue the following command:

```
D UNI, STORAGE
```

The system displays the number of active pages.

To determine the storage that the new conversion image occupies, find the CUN1017I message in the SYSPRINT log that was created in the previous step. This message indicates the number of pages that are required for the new conversion image. For example:

```
CUN1017I GENERATED IMAGE SIZE 291 PAGES........
```

As an alternative, specify a REALSTORAGE value of zero, which indicates that unlimited storage is available. In this case, a value of 524 287 pages is used.

6. Create the PARMLIB member CUNUNIx (PARMLIB member for activating a conversion environment). Normally the member is created in SYS1.PARMLIB, but in this case, you create it in another data set in the logical PARMLIB concatenation. This example uses SYS1.PARMLIB.

The xx can be any two alphanumeric characters, or the special characters @, #, or $. This example uses 00. Here is the sample PARMLIB member in SYS1.PARMLIB(CUNUNI00):

```
REALSTORAGE 292;
IMAGE CUNIMG00;
```

Because CUNMIUTL requires 291 pages, and an additional page is required during IPL, the REALSTORAGE statement indicates that 292 pages of real storage are required.

The IMAGE parameter indicates that the system searches in SYS1.PARMLIB (or a data set in the logical concatenation) member CUNIMG00 for the conversion image.

You can create a PARMLIB member to delete a current conversion environment.

7. Take one of the following actions:
   • Edit IEASYSxx.
     This parameter specifies one or more CUNUNIx PARMLIB members that contain the keywords that configure the conversion environment. Each suffix xx identifies one CUNUNIx member in the PARMLIB concatenation. If several PARMLIB members are specified, they are concatenated in the specified sequence. The concatenated contents is handled internally as a single member. This means that the lines are numbered consecutively, and error messages about syntax errors refer to the concatenated text. Restrictions for keywords apply for the entire concatenated text.
   • Check parameter MAXCAD in IEASYSxx. It limits the amount of common data spaces in a system. If MAXCAD is specified, consider that z/OS support for Unicode creates up to two common data spaces.

8. Initialize the conversion environment with an IPL.

9. After the system is initialized, you can use the DISPLAY UNI system command to show the current z/OS Unicode status or use the SET UNI system command to change the conversion environment.
EBCDIC and ASCII support

Db2 supports both EBCDIC and ASCII encoding of data.

If you specify MIXED DATA = NO on panel DSNTIPF, you can use any compatible SBCS CCSID in the EBCDIC CCSID and ASCII CCSID fields. The first table below lists a selection of common SBCS CCSIDs that might be used as source or target CCSIDs for EBCDIC or ASCII data. Db2 does not support the storing of data into all of these CCSIDs. That is, not all of the numbers listed in the first table below are supported as target CCSIDs in conversion. When you choose CCSIDs, you should choose an EBCDIC CCSID and an ASCII CCSID that are listed in the same row in the first table below.

To determine which combinations Db2 supports with SYSSTRINGS, issue the following SQL statement:

```sql
SELECT * FROM SYSIBM.SYSSTRINGS;
```

If you specify MIXED DATA=NO on installation panel DSNTIPF, specify an SBCS CCSID from the second table below in the EBCDIC CCSID field on DSNTIPF. You must specify a SBCS CCSID from the third table below in the ASCII CCSID field on DSNTIPF. Mixed character data and graphic data cannot be defined on a system when you specify MIXED DATA=NO.

If you specify MIXED DATA=YES on installation panel DSNTIPF, specify a mixed data CCSID from the second table below in the EBCDIC CCSID field on DSNTIPF. You must specify a mixed CCSID from the third table below in the ASCII CCSID field. The second and third tables below show the associated CCSIDs that Db2 assigns for SBCS and DBCS data when you specify a specific MCCSID.

Then specify a mixed CCSID from the third table in the ASCII CCSID field on DSNTIPF. By specifying a CCSID for mixed data (an MCCSID), you also receive system CCSIDs for SBCS and DBCS (graphic) data.

In the second and third tables, four CCSIDs are listed for Japanese to allow for all possible combinations of two single-byte code pages and two double-byte character sets. The difference between the single-byte code pages is in the code points for lowercase Latin letters and Katakana characters. In the code page for Japanese (Extended English, SCSID 1027), lowercase letters have the same code points as other EBCDIC code pages.

In these tables, the terms are used as follows:
**SCCSID**
single-byte coded character set identifier

**MCCSID**
mixed coded character set identifier

**GCCSID**
graphic coded character set identifier

**Table 145. Single-byte coded character set identifiers (CCSIDs)**

<table>
<thead>
<tr>
<th>Country or national language</th>
<th>EBCDIC</th>
<th>ASCII PC</th>
<th>ASCII AIX®</th>
<th>ASCII Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia (U.S. English)</td>
<td>37/1140*</td>
<td>437</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Austria (German)</td>
<td>273/1141*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Belarus (Cyrillic)</td>
<td>1025</td>
<td></td>
<td></td>
<td>1251/5347*</td>
</tr>
<tr>
<td>Belgium</td>
<td>500/1148*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Bosnia and Herzegovina (Cyrillic)</td>
<td>1025</td>
<td></td>
<td></td>
<td>1251/5347*</td>
</tr>
<tr>
<td>Bosnia and Herzegovina (Latin)</td>
<td>870</td>
<td>852</td>
<td>912</td>
<td>1250/5346*</td>
</tr>
<tr>
<td>Brazil (U.S. English)</td>
<td>37/1140*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Bulgaria (Cyrillic Multilingual)</td>
<td>1025</td>
<td></td>
<td></td>
<td>1251/5347*</td>
</tr>
<tr>
<td>Canada (U.S. English)</td>
<td>37/1140*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Croatia</td>
<td>870</td>
<td>852</td>
<td>912</td>
<td>1250/5346*</td>
</tr>
<tr>
<td>Czech Republic</td>
<td>870</td>
<td>852</td>
<td>912</td>
<td>1250/5346*</td>
</tr>
<tr>
<td>Denmark</td>
<td>277/1142*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Finland (Swedish)</td>
<td>278/1143*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>France</td>
<td>297/1147*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Germany</td>
<td>273/1141*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Greece</td>
<td>875 or 423</td>
<td>869</td>
<td>813</td>
<td>1253/5349*</td>
</tr>
<tr>
<td>Hungary</td>
<td>870</td>
<td>852</td>
<td>912</td>
<td>1250/5346*</td>
</tr>
<tr>
<td>Iceland</td>
<td>871/1149*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>International Latin-1</td>
<td>500/1148*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel</td>
<td>424</td>
<td>862</td>
<td>916</td>
<td>1255/5351*</td>
</tr>
<tr>
<td>Italy</td>
<td>280/1144*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Latin America (Spanish)</td>
<td>284/1145*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>FYR Macedonia</td>
<td>1025</td>
<td></td>
<td></td>
<td>1251/5347*</td>
</tr>
<tr>
<td>Netherlands (U.S. English)</td>
<td>37/1140*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>New Zealand (U.S. English)</td>
<td>37/1140*</td>
<td>437</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Norway</td>
<td>277/1142*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
</tbody>
</table>
Table 145. Single-byte coded character set identifiers (CCSIDs) (continued)

<table>
<thead>
<tr>
<th>Country or national language</th>
<th>EBCDIC</th>
<th>ASCII PC</th>
<th>ASCII AIX®</th>
<th>ASCII Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poland</td>
<td>870</td>
<td>852</td>
<td>912</td>
<td>1250/5346*</td>
</tr>
<tr>
<td>Portugal (U.S. English)</td>
<td>37/1140*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Russia (Cyrillic)</td>
<td>1025</td>
<td></td>
<td></td>
<td>1251/5347*</td>
</tr>
<tr>
<td>Serbia and Montenegro (Cyrillic)</td>
<td>1025</td>
<td></td>
<td></td>
<td>1251/5347*</td>
</tr>
<tr>
<td>Serbia and Montenegro (Latin)</td>
<td>870</td>
<td>852</td>
<td>912</td>
<td>1250/5346*</td>
</tr>
<tr>
<td>Spain</td>
<td>284/1145*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Sweden</td>
<td>278/1143*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Switzerland</td>
<td>500/1148*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>Thailand</td>
<td>838</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turkey (Latin 5)</td>
<td>1026</td>
<td>857</td>
<td>920</td>
<td>1254/5350*</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>285/1146*</td>
<td>850/858*</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
<tr>
<td>U.S.A. (U.S. English)</td>
<td>37/1140*</td>
<td>437</td>
<td>819</td>
<td>1252/5348*</td>
</tr>
</tbody>
</table>

Note: * This number represents the equivalent CCSIDs using the euro symbol.

Table 146. EBCDIC double-byte coded character set identifiers (CCSIDs)

<table>
<thead>
<tr>
<th>National language</th>
<th>MCCSID</th>
<th>SCCSID</th>
<th>GCCSID</th>
<th>User-defined characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japanese (Extended Katakana)</td>
<td>930</td>
<td>290</td>
<td>300</td>
<td>4370</td>
</tr>
<tr>
<td>Japanese (Katakana-Kanji)</td>
<td>1390</td>
<td>8482</td>
<td>16684</td>
<td>6205</td>
</tr>
<tr>
<td>Japanese (Extended Katakana)</td>
<td>5026</td>
<td>290</td>
<td>4396</td>
<td>1880</td>
</tr>
<tr>
<td>Japanese (Extended English)</td>
<td>939</td>
<td>1027</td>
<td>300</td>
<td>4370</td>
</tr>
<tr>
<td>Japanese (Latin-Kanji)</td>
<td>1399</td>
<td>5123</td>
<td>16684</td>
<td>6205</td>
</tr>
<tr>
<td>Japanese (Extended English)</td>
<td>5035</td>
<td>1027</td>
<td>4396</td>
<td>1880</td>
</tr>
<tr>
<td>Korean</td>
<td>933</td>
<td>833</td>
<td>834</td>
<td>1880</td>
</tr>
<tr>
<td>Korean</td>
<td>1364</td>
<td>13121</td>
<td>4930</td>
<td>1880</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>935</td>
<td>836</td>
<td>837</td>
<td>1880</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>1388</td>
<td>13124</td>
<td>4933</td>
<td>1880</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>937</td>
<td>28709</td>
<td>835</td>
<td>6204</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>1379</td>
<td>28709</td>
<td>1378</td>
<td>628</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>5475</td>
<td>28709</td>
<td>5474</td>
<td>628</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>1371</td>
<td>1159</td>
<td>9027</td>
<td>628</td>
</tr>
</tbody>
</table>

Chapter 12. Working with international data
Table 147. ASCII double-byte coded character set identifiers (CCSIDs)

<table>
<thead>
<tr>
<th>National language</th>
<th>MCCSID</th>
<th>SCCSID</th>
<th>GCCSID</th>
<th>User-defined characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japanese</td>
<td>932</td>
<td>897</td>
<td>301</td>
<td>1880</td>
</tr>
<tr>
<td>Japanese (Extended)</td>
<td>942</td>
<td>1041*</td>
<td>301</td>
<td>1880</td>
</tr>
<tr>
<td>Japanese (Open environment)</td>
<td>943</td>
<td>1041*</td>
<td>941</td>
<td>1880</td>
</tr>
<tr>
<td>Japanese (HP)</td>
<td>9039</td>
<td>1041*</td>
<td>1351</td>
<td>940</td>
</tr>
<tr>
<td>Korean</td>
<td>949</td>
<td>1088</td>
<td>951</td>
<td>1880</td>
</tr>
<tr>
<td>Korean (EUC)</td>
<td>970</td>
<td>367</td>
<td>971</td>
<td>1880</td>
</tr>
<tr>
<td>Korean</td>
<td>1363</td>
<td>1126</td>
<td>1362</td>
<td>1880</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>1381</td>
<td>1115</td>
<td>1380</td>
<td>1880</td>
</tr>
<tr>
<td>Simplified Chinese (EUC)</td>
<td>1383</td>
<td>367</td>
<td>1382</td>
<td></td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>1386</td>
<td>5210</td>
<td>1385</td>
<td>1880</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>938</td>
<td>904</td>
<td>927</td>
<td>6204</td>
</tr>
<tr>
<td>Traditional Chinese (IBM Big-5)</td>
<td>948</td>
<td>1043</td>
<td>927</td>
<td>6204</td>
</tr>
<tr>
<td>Traditional Chinese (IBM Big-5)</td>
<td>950</td>
<td>1114</td>
<td>947</td>
<td></td>
</tr>
</tbody>
</table>

Note: * The SCCSID 1041 is a superset of SCCSID 897.

Implications of character conversions

Some character conversions have implications that you should be aware of. For example, when a character string is converted to another encoding scheme, the converted string might have a different length from the source string. In this case, the conversion is an expanding conversion or a contracting conversion.

**Expanding conversions:** An expanding conversion occurs when the length of the converted string is greater than that of the source string. For example, an expanding conversion occurs when an ASCII mixed data string that contains DBCS characters is converted to EBCDIC mixed data. Because of the addition of shift characters, an error occurs when an expanding conversion is performed on a fixed-length input host variable that requires conversion from ASCII mixed data to EBCDIC mixed data. The solution is to use a varying-length string variable with a maximum length that is sufficient to contain the expansion. Expanding conversions also can occur when string data is converted to or from Unicode.

**Contracting conversions:** A contracting conversion occurs when the length of the converted string is smaller than that of the source string. For example, a contracting conversion occurs when an EBCDIC mixed data string that contains DBCS characters is converted to ASCII mixed data due to the removal of shift characters. Contracting conversions also can occur when string data is converted to or from Unicode.

Euro symbol support

Db2 enables users to migrate to CCSIDs that support the euro symbol. This support is limited to conversion from specific CCSIDs that do not define the euro symbol to specific CCSIDs that define the euro symbol.
Attention: If you want to migrate to CCSIDs that support the euro symbol, contact IBM Software Support. Altering CCSIDs without specific guidance from IBM Support is strongly discouraged.

The tables below list the ASCII and EBCDIC CCSIDs that can be converted to CCSIDs that support the euro symbol. In most cases, the euro symbol replaces an existing code point, such as the International Currency Symbol (ICS). Unicode UTF-8 (1208) and UTF-16 (1200) support the euro symbol. Unicode SBCS data (367) does not support the euro symbol.

The CCSIDs that you can modify are listed according to the encoding scheme. The first table below lists the EBCDIC CCSIDs, and the second table below lists the ASCII CCSIDs.

**Table 148. EBCDIC CCSID values that convert to euro CCSIDs**

<table>
<thead>
<tr>
<th>CCSIDs without euro symbol</th>
<th>CCSIDs with euro symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>1140</td>
</tr>
<tr>
<td>273</td>
<td>1141</td>
</tr>
<tr>
<td>277</td>
<td>1142</td>
</tr>
<tr>
<td>278</td>
<td>1143</td>
</tr>
<tr>
<td>280</td>
<td>1144</td>
</tr>
<tr>
<td>284</td>
<td>1145</td>
</tr>
<tr>
<td>285</td>
<td>1146</td>
</tr>
<tr>
<td>297</td>
<td>1147</td>
</tr>
<tr>
<td>500</td>
<td>1148</td>
</tr>
<tr>
<td>871</td>
<td>1149</td>
</tr>
</tbody>
</table>

**Table 149. ASCII CCSID values that convert to euro CCSIDs**

<table>
<thead>
<tr>
<th>CCSIDs without euro symbol</th>
<th>CCSIDs with euro symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>850</td>
<td>858</td>
</tr>
<tr>
<td>1250</td>
<td>5346</td>
</tr>
<tr>
<td>1251</td>
<td>5347</td>
</tr>
<tr>
<td>1252</td>
<td>5348</td>
</tr>
<tr>
<td>1253</td>
<td>5349</td>
</tr>
<tr>
<td>1254</td>
<td>5350</td>
</tr>
<tr>
<td>1255</td>
<td>5351</td>
</tr>
<tr>
<td>1256</td>
<td>5352</td>
</tr>
<tr>
<td>1257</td>
<td>5353</td>
</tr>
<tr>
<td>874</td>
<td>4970</td>
</tr>
</tbody>
</table>

You cannot convert other CCSIDs to CCSIDs that support the euro symbol. All databases and all table spaces within an encoding scheme should be altered at the same time.

If you migrate to a euro-enabled CCSID on Db2 and you have Db2 Connect clients, you should set the `DB2CONNECT_ENABLE_EURO_CODEPAGE` environment variable on the client side.
How an entry in SYSIBM.SYSTRINGS works with character conversion

The catalog table SYSIBM.SYSTRINGS contains information about character conversion. Each row in the table describes a conversion from one coded character set to another.

The following columns in SYSTRINGS are relevant to character conversion.

**INCCSID**
The source CCSID of a character conversion.

**OUTCCSID**
The target CCSID of a character conversion.

**TRANSTYPE**
The type of conversion:
- SS: SBCS data to SBCS data
- SM: SBCS data to EBCDIC MIXED data
- MS: EBCDIC MIXED data to SBCS (EBCDIC and ASCII) data
- PS: ASCII MIXED data to SBCS (EBCDIC and ASCII) data
- GG: GRAPHIC data to GRAPHIC data
- PM: ASCII MIXED data to EBCDIC MIXED data
- MM: EBCDIC MIXED data to EBCDIC MIXED data
- MP: EBCDIC MIXED to ASCII MIXED data
- PP: ASCII MIXED to ASCII MIXED data
- SP: SBCS (ASCII and EBCDIC) to ASCII MIXED data

**ERRORBYTE**
Specifies the byte that is used in the conversion table (TRANSTAB) as an error indicator. For example, if ERRORBYTE is X’3E’, that byte is used in the conversion table to indicate that no conversion is defined for code points that map to X’3E’. Null indicates the absence of an error indicator.

**SUBBYTE**
Specifies the byte that is used in the conversion table (TRANSTAB) as a substitution character. For example, if SUBBYTE is X’3F’, that byte is used in the conversion table as a substitute for code points that map to X’3F’. A warning occurs when a code point maps to the value of SUBBYTE. Null indicates the absence of a substitution character.

**TRANSPROC**
The name of a module or a blank string. If IBMREQD is N, a non-blank value of TRANSPROC is the name of a user-provided conversion procedure. If IBMREQD is Y, a non-blank value of TRANSPROC is the name of a Db2 module that contains DBCS conversion tables.

**IBMREQD**
Y indicates that the row is provided by IBM. N indicates that the row has been inserted by the user.

**TRANSTAB**
A 256-byte conversion table or an empty string.

Each row of SYSTRINGS contains information about the conversion of character strings from the coded character set that is identified by INCCSID to the coded...
character set that is identified by OUTCCSID. The conversion function is automatically invoked when a conversion from the coded character set that is identified by the INCCSID column to the coded character set that is identified by the OUTCCSID column is required.

For example, the row of SYSSTRINGS in which the value of INCCSID is 500 and the value of OUTCCSID is 37 describes the conversion from CCSID 500 to CCSID 37. The row in which the value of INCCSID is 37 and the value of OUTCCSID is 500 describes the conversion from CCSID 37 to CCSID 500.

Db2 enforces a distinction between IBM-supplied rows and user-provided rows with the following constraints:

- Rows with IBMREQD=Y cannot be updated or deleted.
- Rows with IBMREQD=N can be inserted, updated, and deleted.
- The same pair of CCSIDs can be in two rows, if one is in an IBM-supplied row and the other is in a user-provided row. In this case, the user-provided row is used for the character conversion.

The following table lists the types of rows that are possible in SYSSTRINGS.

<table>
<thead>
<tr>
<th>The value of TRANSPROC is</th>
<th>The value of TRANSTAB is</th>
<th>The value of IBMREQD is</th>
<th>The result is</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>an empty string</td>
<td>–</td>
<td>No conversion is performed.</td>
</tr>
<tr>
<td>not blank</td>
<td>–</td>
<td>NO</td>
<td>Conversion is performed by the conversion procedure module name identified in the TRANSPROC column</td>
</tr>
<tr>
<td>blank</td>
<td>not empty</td>
<td>–</td>
<td>Conversion is performed by the Db2 module using the conversion table identified in TRANSTAB</td>
</tr>
</tbody>
</table>

Be aware of the following rules for SYSSTRINGS entries:

- An INSERT, UPDATE, DELETE, or LOAD is allowed only if IBMREQD=N.
- The values in the INCCSID and OUTCCSID columns must be in the range 1 - 65533.
- For any given row, the INCCSID and OUTCCSID columns cannot contain the same value.
- The value in the TRANSTYPE column must be SS, SM, MS, PS, MM, PM, GG, MP, PP, or SP.
- For any given row, the ERRORBYTE and SUBBYTE columns cannot contain the same nonnull value.
- The TRANSPROC column must either be blank or contain a string that conforms to the rules for z/OS program names.
- The length that is specified in the TRANSTAB column must be either 0 or 256.

Related reference:

Conversion-related changes that require remote packages to be rebound

Certain conversion-related changes at the local DBMS or at a remote DBMS might force the rebinding of a package.

These include the following changes:

- The system CCSID at the remote DBMS was changed. In this case, always rebind the package.
- The system CCSID at the local DBMS was changed. This could happen, for example, if the wrong system CCSID was specified during installation. If so, string constants in static SQL statements might have been converted incorrectly during the binding of the package. Rebinding corrects the conversion. Other problems might also arise as a result of the change. In most cases, you should rebind the package.
- The subtype of a character column is changed at the remote DBMS. The pertinent changes are from BIT to either SBCS or MIXED, and from SBCS or MIXED to BIT. The change was probably made by modifying the FOREIGNKEY column of the SYSIBM.SYSCOLUMNS table in the remote system catalog. Alternatively, the change might have occurred if the table was dropped and re-created with a different subtype (BIT to either SBCS or MIXED, or either SBC or MIXED to BIT) after the application was bound. A statement that refers to a column with a modified subtype might fail with an SQLCODE of -333. If this occurs, rebind the package containing the statement.

Locales for uppercase and lowercase conversion of EBCDIC data

Rules for uppercase and lowercase usage vary according to language and country. A locale defines the subset of a user's environment that depends on language and cultural conventions.

Db2 uses the information that is associated with a locale to execute UPPER, LOWER, and TRANSLATE functions in a culturally correct manner. A locale consists of two components: the first component represents a specific language and country, and the second component is a CCSID.

Example: In the locale, Fr_CA.IBM-1047, Fr_CA represents the language and country (French Canadian), and IBM-1047 is the associated CCSID.

The symbol for euro currency is supported through the modifier @EURO.

Example: To display results in euro dollars instead of French Francs, specify Fr_FR@EURO.

Db2 uses the SCEELKED and SCEERUN Language Environment (LE) libraries. Both libraries are PDS libraries and are for non-XPLINK linkage only. If you need to customize the locales using Language Environment libraries, see [Internationalization: Locales and Character Sets (XL C/C++ Programming Guide)](#).

The following table shows a partial list of locales that are supplied with z/OS C/C++. For a more complete list of locales, see [Compiled locales (LE locales) (XL C/C++ Programming Guide)](#).
### Setting up z/OS to allow uppercase and lowercase conversion of Unicode and ASCII data

If you want to use the UPPER or LOWER built-in functions to process Unicode and ASCII data, you must complete additional setup. You must also complete additional setup if you want to use uppercase or lowercase conversions for characters other than A-Z or a-z.

**Procedure**

To allow uppercase and lowercase conversion of Unicode and ASCII data:

Add control statements to the configuration of z/OS Support for Unicode services.

**Example**

Assume that your EBCDIC CCSID is 37 and your ASCII CCSID is 819, and that you have defined the following conversions:

```
CONVERSION 37,367,ER;
CONVERSION 37,1208,ER;
CONVERSION 37,1200,ER;
CONVERSION 367,37,ER;
CONVERSION 1208,37,ER;
CONVERSION 1200,37,ER;
CONVERSION 819,367,ER;
CONVERSION 819,1208,ER;
CONVERSION 819,1200,ER;
CONVERSION 367,819,ER;
CONVERSION 1208,819,ER;
CONVERSION 1200,819,ER;
```

Modify your conversion image as follows:

```
CASE NORMAL; /* normal casing */
CASE SPECIAL; /* additional locale-independent casing */
CASE LOCALE; /* additional locale-dependent casing */
```
The three additional CASE statements provide the necessary infrastructure for the UPPER and LOWER functions to process Unicode data according to the Unicode Standard.
Information resources for Db2 12 for z/OS and related products

Information about Db2 12 for z/OS and products that you might use in conjunction with Db2 12 is available online in IBM Knowledge Center or on library websites.

Obtaining Db2 for z/OS publications

Current Db2 12 for z/OS publications are available from the following websites:


Links to IBM Knowledge Center and the PDF version of each publication are provided.

Db2 for z/OS publications are also available for download from the IBM Publications Center [http://www.ibm.com/shop/publications/order].

In addition, books for Db2 for z/OS are available on a CD-ROM that is included with your product shipment:

• Db2 12 for z/OS Licensed Library Collection, LKxT-xxxx, in English. The CD-ROM contains the collection of books for Db2 12 for z/OS in PDF format. Periodically, IBM refreshes the books on subsequent editions of this CD-ROM.

Installable information center

You can download or order an installable version of the Information Management Software for z/OS Solutions Information Center, which includes information about Db2 12 for z/OS, QMF, IMS, and many Db2 Tools for z/OS products. You can install this information center on a local system or on an intranet server. For more information, see [http://www-01.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/com.ibm.db2z11.doc/src/alltoc/installabledzic.html](http://www-01.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/com.ibm.db2z11.doc/src/alltoc/installabledzic.html).
Notices

This information was developed for products and services offered in the US. This material might 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, MD-NC119
Armonk, NY 10504-1785 US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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, MD-NC119
Armonk, NY 10504-1785 US

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 document 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.

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 actual people or business enterprises 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 shown below:

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. (enter the year or years).

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This information is intended to help you to plan for and administer Db2 12 for z/OS. This information also documents General-use Programming Interface and Associated Guidance Information and Product-sensitive Programming Interface and Associated Guidance Information provided by Db2 12 for z/OS.
General-use Programming Interface and Associated Guidance Information

General-use Programming Interfaces allow the customer to write programs that obtain the services of Db2 12 for z/OS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

---

Product-sensitive Programming Interface and Associated Guidance Information

Product-sensitive Programming Interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive Programming Interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

---

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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.shtml](http://www.ibm.com/legal/copytrade.shtml).

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

**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 IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.
Glossary

The glossary is available in IBM Knowledge Center.

See the [Glossary] topic for definitions of Db2 for z/OS terms.
Index

Special characters

SVOLARC subsystem parameter 293

Numerics

0268 636
32K SOR PRIMARY SPACE field of panel DSNTIP9 155
32K SOR WORK SEG SIZE field of panel DSNTIP9 156
32K SOR WORK TS'S field of panel DSNTIP9 156
32K TEMP PRIMARY SPACE field of panel DSNTIP9 158
32K TEMP WORK SEG SIZE field of panel DSNTIP9 159
32K TEMP WORK TS'S field of panel DSNTIP9 159
4K SOR PRIMARY SPACE field of panel DSNTIP9 154
4K SOR WORK SEG SIZE field of panel DSNTIP9 155
4K SOR WORK TS'S field of panel DSNTIP9 155
4K TEMP PRIMARY SPACE field of panel DSNTIP9 156
4K TEMP WORK SEG SIZE field of panel DSNTIP9 157
4K TEMP WORK TS'S field of panel DSNTIP9 157

A

ABIND subsystem parameter 184
ACBNAME option of APPL statement 891
ACCEL subsystem parameter 226, 374
ACCELERATION OPTIONS field of panel DSNTIP82 227
ACCELERATION STARTUP field of panel DSNTIP82 226
accelerators
database objects
creating 936
migrating 937
ACCEPT job 44
ACCESS CONTROL field of panel DSNTIPO3 190
accessibility
keyboard xiii
shortcut keys xiii
ACCUMACC subsystem parameter 180
ACCUMUID subsystem parameter 181
active log
data set
prefix 112
storage requirements 11
dual logging 109
installation 422
size
estimating 11, 135
adding
data sharing members 682
employee information at remote locations 817
adding a data sharing member
defining address space startup procedures 685
system procedure libraries 685
adding data sharing members
CLIST 683
connecting CICS 691
user authorization exit routines 688
additional capabilities
for Db2 829
address spaces
administrative task scheduler
storage requirements 21
allied agent 20
database services
description 20
working calculation 31
working storage calculation 32
DDF 19
IRLM 19
system services 20
WLM-established stored procedures 21
WLM-managed 835
ADMIN SCHEDULER field of panel DSNTIPX 312
administrative enablement routines
enabling 851
administrative task scheduler
address space 470
storage requirements 21
address space startup procedure 470
routines 470
enabling 851
set up 470
table spaces 470
ADMINPROC subsystem parameter 374
ADVANCED CONFIGURATION OPTIONS field of panel DSNTIP1 322
advantages
non-volatile coupling facility 632
AGENT LEVEL THRESHOLD field of panel DSNTIP91 162
AGGREGATION FIELDS field of panel DSNTIPN 181
alias
server location 885
allied agent address space 20
ALTER GROUPBUFFERPOOL 645
ALTERNATE COPY POOL field of panel DSNTIP62 256
ALTERNATE_CP subsystem parameter 256
AMDPRECTL module 415
APN option of DSNMPN macro 876, 878
APPC option of APPL statement 890
APPL COMPAT LEVEL field of panel DSNTIP41 94
APPL DEFAULTS field of panel DSNTIP03 189
APPL REGISTRATION TABLE field of panel DSNTIPZ 348
APPL statement 887, 901
example 887
options
description 887
DSESCLIM 908
LUNAME 916
APPLCOMPAT subsystem parameter 94
application defaults load module
installation 424
application defaults parameters 667
APPLICATION ENCODING field of panel DSNTIPF 203
application environment
WLM 835
APPLICATION LOAD field of panel DSNTIPT 116
archive log
cataloging options 78
data set
prefix 113

© Copyright IBM Corp. 1982, 2017

967
archive log (continued)  
data set (continued)  
  storage requirements 18  
dual logging 109  
ARCHIVE LOG FREQ field of panel DSNTIPL 279  
ARCHIVE LOG RACF field of panel DSNTIPP 261  
ART/ORT ESCAPE CHARACTER field of panel DSNTIPZ 547  
ASCII  
  case conversion 955  
  conversion table 947  
ASCII CODED CHARACTER SET field of panel DSNTIPF 202  
asynchronous duplexing  
  lock structure 636  
ATCSTRxx member of SYS1.VTAMLST library 899  
attachment facilities  
  CICS 437  
    adding data sharing members 715  
    installation 437  
    migration 492  
  IMS  
    adding data sharing members 690  
    installation 437, 871  
    migrating data sharing members 714  
    migration 492  
  TSO 430  
    connecting for data sharing 677  
    installation 430  
    migration 489  
AUDIT TRACE field of panel DSNTIPN 176  
AUTH EXIT CACHE REFER field of panel DSNTIPP 264  
AUTH EXIT CHECK field of panel DSNTIPP 264  
AUTH EXIT LIMIT field of panel DSNTIPP 263  
AUTH MEMBER field of panel DSNTIPM 276  
AUTH option  
  APPL statement 890  
  AUTH SEQUENCE field of panel DSNTIPM 276  
AUTHID column  
  MODESELECT table 904  
  USERNAMES catalog table 926  
authorization  
  installation SYSPR 6  
  SYSADM 6  
authorization ID  
  RACF 429  
  TSO 429  
AUTO BIND field of panel DSNTIPO 184, 561  
AUTO START field of panel DSNTIPI 234  
automatic  
  rebuild of coupling facility structures 630  
  recall 184  
automatic rebinds  
  considerations in a coexistence environment 707  
automatic recovery 629  
automatic restart 625  
  advantage 625  
  policy 625  
  creating 626  
AUTOREC (YES) 629  
AUTOSES option of APPL statement 888  
availability  
  catalog and directory 637  
  coupling facility 627  
  planning 624  
  volatility of coupling facility 632  

B  
BACKODUR subsystem parameter 284  
BACKOUT DURATION field of panel DSNTIPL 284  
backup  
  migration 495  
BIF COMPATIBILITY field of panel DSNTIPX 310  
BIF_COMPATIBILITY subsystem parameter 310  
BIND NEW PACKAGE field of panel DSNTIPP 271  
  binding  
    installation 443  
    migration  
      DCLGEN 509  
      SPUFI 509  
    recommendations for a coexistence environment 708  
    remote package  
      plan name for 904  
      relation to character conversion 954  
BLOCK SIZE field of panel DSNTIPA 289  
bootstrap data sets (BSDS) 497  
BP0 - BP29 fields of panel DSNTIPL 173  
BP30 - BP32K9 fields of panel DSNTIPP 175  
BSDS (bootstrap data set)  
  adding second 497  
  increasing size 655  
  installation 422  
  naming recommendation 663  
  storage requirements 14  
BSDS, extended format  
  installing Db2 482, 483  
BUFFER POOL SIZE field of panel DSNTIPC 355  
buffer pools  
  fallback implications 559  
  group sizes 641  
  storage requirement 25  
  VTAM IOBUF storage requirements 907  

C  
C/LE PRELINKER PHASE field of panel DSNTIPU 126  
C/370 COMPILER MESSAGES field of panel DSNTIPL 126  
C/CPP COMPILER LIBRARY field of panel DSNTIPL 125  
C/CPP COMPILER MODULE field of panel DSNTIPL 125  
CACHE DYN STABILITY field of panel DSNTIP8 213  
CACHE DYNAMIC SQL field of panel DSNTIPF 212  
CACHE option of CREATE SEQUENCE statement 656  
cache sequencing 656  
cache structure 619  
cache structure sizes 646  
CACHE_DEP_TRACK_STOR_LIM subsystem parameter 374  
CACHEDYN_FREELocal subsystem parameter 374  
CACHEDYN_STABILIZATION subsystem parameter 213  
caching all data 644  
CAF (call attachment facility)  
  coexistence considerations 707  
catalog alias 611  
CATALOG ALIAS field of panel DSNTIPA2 78  
catalog level 705  
catalog tables  
  SYSTRINGS  
    description 952  
catalog, DB2  
  checking 508  
  installing 422  
  storage requirements 10  
catalog, integrated catalog facility 421  
CATTDDLACL subsystem parameter 81
CHKTYPE field of panel DSNTIPL1 281

CHKTYPES subsystem parameter 281

CICS
adding data sharing members 715
applications, migration considerations in a Sysplex 657
CICSPlex SM 657
connecting to Db2 691
adding data sharing members 691
installing 437
preparing for data sharing 653
migration 492
operating
starting application 768
testing 765
reconnecting 557
storage handling facilities 769
CICS COBOL LIBRARY field of panel DSNTIPW 134
CICS EXCI LIBRARY field of panel DSNTIPW 134
CICS LOAD LIBRARY field of panel DSNTIPW 134
CICS MACRO LIBRARY field of panel DSNTIPW 134
CICS PL/I LIBRARY field of panel DSNTIPW 134
CICS transaction invocation stored procedure
creating 854
class of service modifying 902
CLIST
adding data sharing members 683
calculations 9
enabling data sharing 669
jobs tailored by 360
migrating data sharing members 714
tailoring installation jobs 363
update process 367
CLIST ALLOCATION field of panel DSNTIPA3 86
CLIST LIBRARY field of panel DSNTIP 116
CLIST messages fields of panel DSNTIPC 357
CLIST messages fields of panel DSNTIPC1 358
CNOS (change number of sessions)
description 908
COBOL COMPILER LIBRARY field of panel DSNTIPU 129
COBOL LE PRELINKER PHASE field of panel DSNTIPU 129
coded character set 941
coeexistence of Db2 releases 36
WLM-managed stored procedures 704
coeexistence of DB2 releases
IRLM service 710
COLUMNS field of panel DSNTIPD 136
command line processor
setting up 935
COMMAND PREFIX field of panel DSNTIPM 274
command recognition character (CRC) 872
COMMAND SCOPE field of panel DSNTIPM 277
commands
prefix
message format 654
common service area (CSA) 18, 22
calculating storage requirements 23
common SQL API stored procedures enabling 855
communications database (CDB) 879
COMPACT DATA field of panel DSNTIPA 293
COMPRESS DB2 DIR LOBS field of panel DSNTIPA 82
COMPRESS SMF RECS field of panel DSNTIPN 183
COMPRESS SPT01 field of panel DSNTIPA 82
COMPRESS_DIRLOB subsystem parameter 82
COMPRESS_SPT01 subsystem parameter 82
compression
archive log data set 293

CATDMGCL subsystem parameter 81
CATDSTCL subsystem parameter 81
CATMAINT utility
migration from Db2 11 440, 507
CATXDLACL subsystem parameter 82
CATXMGCL subsystem parameter 82
CATXSTCL subsystem parameter 82
CCSID (coded character set identifier)
code page 947
definition 942
installation option 201
specifying 942
CDB (communications database)
creating during installation 443
description 895, 926
dropping while DDF is active 906
IPLIST table 926
IPNAMES table 926
LOCATIONS table 879, 895, 898, 926
LULIST table 895
LUMODES table 898
LUNAMES table 895
MODESELECT table 898
populating
connecting Db2 subsystems 926
populating while
connecting Db2 subsystems 895
installing 469
updating while DDF is active 906
USERNAMES table 895, 926
CDDS_MODE in macro DSN6LOGP 376
CDDS_PREF in macro DSN6LOGP 377
CDSSRDEF subsystem parameter 222
CFRM policy 633, 634
description 604
REBUILDPERCENT parameter 631
CFSizer tool 640
change log inventory utility
changing
generic LU name 916
location name 916
LU name 916
VTAM password 916
character conversion
code page 947
code point 947
contracting conversion 950
description 941
distributed data 941
Euro symbol 951
expanding conversion 950
locale
definition 954
specifying 954
support for euro currency 954
lowercase 955
rebinding remote packages 954
SYSIBM.SYSTRINGS catalog table 952
Unicode 942
setup 943
upercase 955
character strings
transmitting 941
CHECK INDEX utility
migration preparation 485
CHECK_FASTREPLICATION subsystem parameter 378
CHKTYPES field of panel DSNTIPL1 281

Index 969
COMPROT option of MODEENT macro 893
configuration
disk connectivity 607
Parallel Sysplex components 602
configuration control statements 846
CONN QUEUE MAX DEPTH field of panel DSNTIP5 306
CONN QUEUE MAX WAIT field of panel DSNTIP5 306
connection
database management systems 879
Db2 895
systems 879
DB2I panels to ISPF installation 433
migration 491
connection exit routine
installation 426
migration 503
connectivity failures 630
connectivity checking 607
consumer
web service 865
contention
for conversation 888
contention, false
LOCK ENTRY SIZE parameter of installation panel DSNTIP1 648
CONTROL ALL APPLICATIONS field of panel DSNTIP3 347
conversation
contention 888
definition 882
SQL 903
system 895
CONVLIMIT column of LUMODES catalog table
CNOS negotiation 908
description 903
COPY 1 NAME field of panel DSNTIP1 110
COPY 1 PREFIX field of panel DSNTIP1 111, 112
COPY 2 NAME field of panel DSNTIP1 110
COPY 2 PREFIX field of panel DSNTIP1 112, 113
COPY FAST REPLICATION field of panel DSNTIP61 249
COPY field of panel DSNTIP61 252
COPY_FASTREPLICATION subsystem parameter 249
coupling facility
availability 627
channel failure 630
failure-isolated 627
failures 628
lock name 619
non-failure-isolated 627
non-volatile 632
placement 627
relation to INITSIZE 641
storage
  group buffer pool 641
  lock 647
  SCA 649
structure sizer 640
structures
  allocation size 640
  changing size of 640, 650
  estimating storage 639
  policy definition 604
types 602
  volatile 632
CPP AUTO CALL LIBRARY field of panel DSNTIPU 127
CPP CLASS LIB HEADERS field of panel DSNTIPU 127
CPP CLASS LIBRARY field of panel DSNTIPU 127
CPP LE PRELINKER PHASE field of panel DSNTIPU 127
CPP PROCEDURE LIBRARY field of panel DSNTIPU 128
CRC (command recognition character)
  installation 872
CREATE SEQUENCE statement 656
critical data 657
CROSS MEMORY
  IRLM 19
cross-system coupling facility (XCF) 602
CSA (common service area) 22
CTC (channel-to-channel)
  considerations for MAXBFRU size 912
  sample definitions for VTAM 912
  X’ sense code 912
CURRENT DEGREE field of panel DSNTIP81 222
CURRENT MAINT TYPES field of panel DSNTIP81 225
CURRENT QUERY ACCEL field of panel DSNTIP82 230
CURRENT REFRESH ACCEL field of panel DSNTIP81 225

D
data
  compression 825
distributed 879
data compression
  Huffman 825
DATA DEF TIMEOUT field of panel DSNTIPE1 169
DATA SET NAME PREFIX field of panel DSNTIP1 71
DATA SET NAME SUFFIX field of panel DSNTIP1 71
DATA SET(MEMBER) NAME field of panel DSNTIP1 69
data sets
  altering for data sharing 674
  control block size calculation 30
  defining for new members 686
  storage requirements 10
data sharing
  adding members 682
    creating an image copy 694
    default storage group 693
    defining Db2 to z/OS 691
    defining initialization parameters 687
    defining system data sets 686
    recording data to SMF 688
    REXX Language Support 693
    running the CLIST 683
    starting Db2 691
temporary work files 693
  verifying subsystem 694
application design considerations 656
  catalog merging 695
disablement 728
disabling 728
enabling 668
  altering active log data sets 674
  connecting to TSO 677
  creating an image copy 681
default storage group 680
  defining Db2 to z/OS 678
defining initialization parameters 674
  introduction 653
  recording data to SMF 676
  running the CLIST 669
  starting Db2 678
  stopping Db2 670
temporary work files 680
  verifying subsystem 682
data sharing (continued)
migrating
    stopping Db2  715
    migrating groups
        first member  713
        preparation  711
        subsequent members  714
    migrating members
        running the CLIST  714
        migrating transactions  660
    Parallel Sysplex  602
    procedures  601
    re-enablement  728
    re-enabling  731
    release coexistence  703
    starting inactive members  724
DATA SHARING field of panel DSNTIPA1  68
data sharing group
    creating  661
database
    DSNDBo6 (Db2 catalog database)  422
    DSNDBo7 (work file database)  441
database descriptors (DBDs)  651
database services address space  20
DATABASES field of panel DSNTIPD  136
DATABASES field of panel DSNTPE  164
DATASET STATS TIME field of panel DSNTIPN  179
DATE FORMAT field of panel DSNTIP4  208
DB BACKUP STG GROUP field of panel DSNTIP62  257
Db2
    hardware requirements  5
    software requirements  5
Db2 12
    migrating  476
    Db2 12 conversion mode
        verifying  537
    Db2 12 IVP objects
        verifying migration  537
    DB2 Connect
        creating database during installation  445
DB2 GENERIC LUNAME
    field of panel DSNTIPR  885
DB2 GENERIC LUNAME field of panel DSNTIPR  297
DB2 LOCATION NAME field of panel DSNTIPR  296
DB2 TCP/IP IPNAME field of panel DSNTIPR  298
DB2 VTAM LUNAME field of panel DSNTIPR  297
DB2 VTAM PASSWORD field of panel DSNTIPR  297
DB2 web services
    enabling  860
    zHyperLinks SCOPE field of panel DSNTIPA3  85
Db2-supplied
    stored procedures  829
    user-defined functions  829
Db2-supplied routines  829
DB2 (Db2 Interactive)
    connecting to  491
    connecting to ISPF
        installation  433
    panels
        verifying values  481
DBACKWV subsystem parameter  272
DBADM CREATE AUTH field of panel DSNTIP1  272
DBCS (double-byte character set)
    identifiers  947
DBRM LIBRARY field of panel DSNTIP  116, 118
DBRMLIB DATA library data set
    DASD volume  86
DBRMLIB DATA library data set (continued)
    device type  84
    DSNTIJIN job  422
    naming considerations  42
    DCLGEN (declarations generator)
        migration  509
    DDCS (data definition control support)
        creating database during installation  445
    DDF (distributed data facility)
        address space  19
        installation
            implications  881
        overview  879
    DDF STARTUP OPTION field of panel DSNTIPR  296
    DDF THREADS field of panel DSNTIPR  300
    DDF_COMPATIBILITY subsystem parameter  374
    DDF/RSAF ACCUM field of panel DSNTIPN  180
    DDL registration
        storage estimation  18
    DDLTOX subsystem parameter  169
    DDRAINL option of APL statement  891
    DEADLOCK CYCLE field of panel DSNTIPJ  241
    DEADLOCK TIME field of panel DSNTIPJ  241
    DEALLOC PERIOD field of panel DSNTIPA  290
    DECFLIGHT ROUNDDING MODE field of panel DSNTIPF  204
    DECIMAL ARITHMETIC field of panel DSNTIP4  206
    DECIMAL POINT IS field of panel DSNTIPF  198
    DECLARATION LIBRARY field of panel DSNTIP7  117
    DECPSSID parameter of DSNHDECP  667
    DEF ENCODING SCHEME field of panel DSNTIPF  203
    DEFAULT 16-KB BUFFER POOL FOR USER DATA field of panel DSNTIP1  172
    DEFAULT 32-KB BUFFER POOL FOR USER DATA field of panel DSNTIP1  172
    DEFAULT 4-KB BUFFER POOL FOR USER DATA field of panel DSNTIP1  171
    DEFAULT 8-KB BUFFER POOL FOR USER DATA field of panel DSNTIP1  172
    DEFAULT BUFFER POOL FOR USER INDEXES field of panel DSNTIP1  173
    DEFAULT BUFFER POOL FOR USER LOB DATA field of panel DSNTIP1  173
    DEFAULT BUFFER POOL FOR USER XML DATA field of panel DSNTIP1  173
    default database (DSNDBo4)
        storage estimation  17
    DEFAULT PARTITION SEGSIZE field of panel DSNTIP71  144
default storage group
    adding data sharing members  693
    enabling data sharing  680
    installing Db2  441
    DEFAULT TEMPLATE field of panel DSNTIP61  252
    DEFAULT_INSERT_ALGORITHM subsystem parameter  148
    DEFER field of panel DSNTIPS  294
    DEFINE CATALOG field of panel DSNTIP2  80
    DEFINE DATA SETS field of panel DSNTIP71  149
    defining initialization parameters
        adding data sharing members  687
        enabling data sharing  674
    DELCF STRUCTS field of panel DSNTIPK  99
    DEL_CFSTRUCTS_ON_RESTART subsystem parameter  99
    deleting
        data sets, DSNTIJEDE  422
    DESCRIBE FOR STATIC field of panel DSNTIP4  207
    DESCSTAT subsystem parameter  207
    DEVICE TYPE 1 field of panel DSNTIPA  288
DEVICE TYPE 2 field of panel DSNTIPA 289
DFSIEL DD statement 40, 872
DFSMS (Data Facility Storage Management Subsystem) installation 261
DFSMSshm (Data Facility Hierarchical Storage Manager) RECALL command 184
DFSORT (Data Facility Sort) program library 415
directory installing 422
storage requirements 10
DIRECTORY AND CATALOG DATA field of panel
DSNTIPA2 81
DIRECTORY AND CATALOG INDEXES field of panel
DSNTIPA2 82
disability xiii
DISABLE EDM RTS field 218
DISABLE_EDMRTS subsystem parameter 218
disabling
data sharing 728
data recovery 731
DISCONNECT IRLM field of panel DSNTIPJ 243
disk, shared requirement 607
DISPLAY GROUP command
group and member release level 705
DISPLAY NET,BFRUSE command of VTAM 899
DIST SQL STR DELIMTR field of panel DSNTIPF 200
distributed data
moving to data sharing 696
planning
DRDA access 879
number of systems that can be connected 879
programming
character conversion 941
distributed organization application
DRDA access 812
distributed unit of work 879
distribution libraries
manage use DFSMSshm 184
SMF/E 44
DL/I BATCH TIMEOUT field of panel DSNTIPI 237
DMINWNL option of APPL statement 888
DMINWNR option of APPL statement 888
domain name
definition 918
domain name server
definition 918
DPROP SUPPORT field of panel DSNTIPO 185
DPEGSZ subsystem parameter 144
DRDA (Distributed Relational Database Architecture) remote access control 881
DRDA access
definition 881
distributed organization application 812
setting up 879
specifying modes 903
updating 814
DRDA RDBNAM (relational database name) 885
DRESPL option of APPL statement 891
DSESILM option of APPL statement
CNO5 negotiation 908
description 888
DSMAX field of panel DSNTIPC 351
DSMAX limit on open data sets, description 30
DSMAX subsystem parameter 351
DSNICOPY utility merging Db2 subsystems 696
DSNICOPY utility (continued)
migration preparation 485
DSN1DIST address space 19
DSN3ATH connection exit routine 426
adding data sharing members 688
DSN3SGN sign-on exit routine
adding data sharing members 688
DSN6ARVP macro 424
DSN6FAC macro 424
DSN6GRP macro 424
DSN6LOGP macro 424
DSN6SPRM macro
installation 424
DSN6SYSF macro 424
DSN8EAEL exit routine 824
DSN8HUFF edit routine 825
DSNACEPJ job 44
DSNACICS stored procedure
creating 854
DSNADMT startup procedure 470
DSNALLOC job 44
DSNAPPL1 job 44
DSNBIND plan name in SYSIBM.MODESELECT table 904
DSNC STR command 653
DSNCONNS ISPF table 432
DSNDB06 database
installation job DSNTIJIN 422
DSNDBRMS ISPF table 432
DSNDFD database 879
DSNEMCO1 CLIST 431
DSNEPR1 panel of ISPF 433
DSNETBLS data set for ISPF tables 432
DSNHDECP load module
installation 424
list of parameters 396
migration 495, 717
shared 667
DSNU03 job 925
DSNMAPN macro 876
DSNPLPKN ISPF table 432
DSNRECV1 job 44
DSNRECV3 job 44
DSNTEJ1 job 726
DSNTEJxx job
installation 734
migration 537
data sharing members 721
DSNTESA job 755
DSNTESC job 755
DSNTESD data set member 726
DSNTESA job 755
DSNTESQ queries 488, 827
DSNTIAD sample program
executes SQL statements 827
invoked by DSNTJE1 743
run by DSNTIJTM 441, 509
DSNTIDXA member 72
DSNTIDXB member 72
DSNTIDxx member 73
DSNTIJAM job 937
DSNTJAS job 936
DSNTJCA job 421
DSNTJCB job 484, 725
DSNTJCVC job 482
DSNTIJDE job 422, 423
adding data sharing members 686
H

HAVAIL option of APPL statement 890
Hierarchical Storage Manager (DFSMShsm) 184
HIGH LEVEL ASSEMBLER LIB field of panel DSNTIPU 124
HONOR_KEEPDICTIONARY subsystem parameter 374
HSM MESSAGE DS HLQ field of panel DSNTIP62 258
HTTP proxy support
  - REST user-defined functions 866
  - Huffman compression
    - exit routine 825
  - HVSHARE parameter 22

I

IBM BigInsights user-defined functions
  - creating 869
  - IBM Data Server Driver
    - installing 871
IBM InfoSphere BigInsights user-defined functions 869
IBM LE LINK EDIT LIB field of panel DSNTIPU 124
IBM LE PRELINK MSG LIB field of panel DSNTIPU 124
IBM LE RUNTIME LIBRARY 2 field of panel DSNTIPU 123
IBM LE RUNTIME LIBRARY field of panel DSNTIPU 123
IBM Text Search for Db2 for z/OS
  - enabling 936
IBMDB2LM mode 892
IBMRRDB mode 892
IBMUSER authority, establishing authorization ID 429
IDENTIFY/AUTH field of panel DSNTIP03 190
IDLE THREAD TIMEOUT field of panel DSNTIP5 302
IDRC feature 293
IEAAPFx member of SYS1.PARMLIB
  - DSNTIJMV job
    - installation 415
  - DSNTIPM panel 273
IEFSSNx member of SYS1.PARMLIB
  - DSNTIJMV job
    - installation 415
  - DSNTIPM panel 273
IEFSNxx parmlib member
  - command prefix 654
IFCID (instrumentation facility component identifier)
  - identifiers by number
    - 0267 632
    - 0268 632
  - IFCIDs 0267 636
IGNORE SORTNUM STAT field of panel DSNTIP6 246
image copy
  - adding data sharing members 694
  - data sharing 681
IMMEDIATE WRITE field of panel DSNTIP8 216
IMPDSDEF subsystem parameter 149
IMPDSSIZE subsystem parameter 374
IMPLICIT TIME ZONE field of panel DSNTIP4 210
IMPTKMOD subsystem parameter 374
IMPTSCMP subsystem parameter 150
IMS BMP TIMEOUT field of panel DSNTPI 237
IMS connection 871
  - adding data sharing members 690
  - data sharing 654
  - defining
    - new programs 876
    - new transactions 876
  - installation 437, 871
  - migrating data sharing members 714
  - migration 492
  - operating
    - starting 764
IMS connection (continued)
operating (continued)
testing 760
reconnecting 557
IMS language interface module (DFSLI000) generating 877
IMS RESLIB field of panel DSNTIPW 133
IMS.PROCLIB library description 872
IMSRCIL macro 872
INCLUDE LIBRARY field of panel DSNTIP 119
incompatibilities of releases 481
INDEX CLEANUP THREADS field of panel DSNTIPE1 169
INDEX SPACE ALLOCATION field of panel DSNTIP7 142
INDEX_CLEANUP_THREADS subsystem parameter 169
INDEX.IO_PARALLELISM subsystem parameter 374
indexes
type 1 478
initial program load (IPL) 437
adding data sharing members 691
enabling data sharing 678
INITSIZE parameter of CFRM policy relation to SIZE 641
INLISTP subsystem parameter 374
INPUT MEMBER NAME field of panel DSNTIPA1 72
input variable files
z/OSMF samples 593
INSTALL DD CONTROL SUPT field of panel DSNTIPZ 346
INSTALL DS PREFIX field of panel DSNTIPG 106
INSTALL GRANTEE(S) field of panel DSNTIPG 104
INSTALL IC PREFIX field of panel DSNTIPG 106
INSTALL IRLM field of panel DSNTIP 233
INSTALL PKG OWNER field of panel DSNTIPG 103
INSTALL SQL ID field of panel DSNTIPF 102
INSTALL TYPE field of panel DSNTIP1A 67
installation
choosing options to enable data sharing 661
defining DB2 to IRLM and MVS 415
description 1
IRLM 232, 238
macros
DSN6ARVP 424
DSN6FAC 424
DSN6GRP 424
DSN6LOCVP 424
DSN6PRM 424
DSN6SYSP 424
output 358
overview 1
preparing 5
tailoring session 1
tapes or cartridges 37
verification 473
buffer pool caching for data sharing 726
concurrency for data sharing 727
for data sharing 726
global lock serialization for data sharing 727
planning 739
testing batch environment 748
testing CICS environment 765
testing IMS environment 760
testing PL/I batch 755
testing SPUFI 755
testing TSO attachment facility 743
verifying 733
installation CLIST (continued)
calculating disk and storage requirements 9
disk requirements calculating 9
enabling data sharing 669
jobs tailored by 360
migrating data sharing members 714
migration 1
update process 367
installation jobs
description 1
DSNTIJC 421
DSNTIJDE 422
DSNTIJEX 426
DSNTIJIC 469
DSNTIJJD 425
adding data sharing members 688
DSNTIJIE 425
DSNTIJIF 425
DSNTIJIN 422
DSNTIJMV
installation 415
migration 498
DSNTIJSC 443, 444
DSNTIJTM 441, 509
adding data sharing members 693
enabling data sharing 680
DSNTIJUA 424
DSNTIJUL 424
DSNTIJUM 424
DSNTIJUZ 424
DSNTIJVC 431
system affinity 348
tailoring 363
installation libraries
performance implications 41
installation panels
DSNTIPC 9
DSNTIPD 9
DSNTIPRI 318
DSNTIPRA 322
DSNTIPRB 323
DSNTIPRC 324
DSNTIPRE 325
DSNTIPRF 327
DSNTIPRH 328
DSNTIPRI 330
DSNTIPRJ 332
DSNTIPRK 333
DSNTIPRL 335
DSNTIPRM 339
DSNTIPRN 340
DSNTIPRO 341
DSNTIPRP 343
DSNTIPRQ 344
installation SYSADM authority
authorization IDs established 429
installation SYSSOPR 6
installation SYSSOPR authority
authorization IDs established 429
installation verification procedure (IVP) 559
adding data sharing members 694
enabling data sharing 682
installation verification samples
OBA stored procedure 785
stored procedure with result set 776
stored procedure without result set 774
976 Installation and Migration Guide
IP addresses
definition 918
IPADDR column of IPLIST catalog table 926
IPADDR column of IPNAMES catalog table 926
IPL (initial program load)
  adding data sharing members 691
  enabling data sharing 678
  installation 437
IPv4 932
IPv6 932
IRLM
defining Db2 to 415
fallback, stopping IRLM during 554
group naming, naming convention 96
libraries
  load 114
  sample 44
  target 44
naming convention for group names 96
space
  estimating maximum 350
SYS1.PARMLIB updates 415
z/OS system linkage index 415
IRLM (internal resource lock manager)
  address space 19
  automatic restart 625, 626
  automatic start of diagnostic traces 663
coeexistence 706
coupling facility lock structure size 647
dump formatting module, AMDPRECT 415
every in PPT 415
function level 706
LOCK ENTRY SIZE parameter of installation panel DSNTIPJ 648
lock storage 651
monitoring storage 652
names 618, 626
priority 652
starting
  adding data sharing members 691
  after fallback 557
  during data sharing migration 719
  during migration 505
  enabling data sharing 678
  installation 438
subsystem parameters 663
IRLM LOAD LIBRARY field of panel DSNTIPT 118
IRLM XCF GROUP NAME field of panel DSNTIPJ 242
IRLMAUT subsystem parameter 234
IRLMPROC 19
IRLMRTW subsystem parameter 233
IRLMSID subsystem parameter 233
ISPF
  migration 491
  ISPF (Interactive System Productivity Facility)
    primary option menu, panel connection 433
  ISPF ISPLINK MODULE field of panel DSNTIPW 133
  INSTINCLM mode table 888, 892
IVP (installation verification procedure)
  adding data sharing members 694
  C language options 740
  C++ language options 740
  COBOL options 739
  enabling data sharing 682
  fallback 559
  installation 733
  migration 733
  phases 734
  PL/I options 741
  preparing for 481
  programs-phases relationship 734
IVP DATA LIBRARY field of panel DSNTIPG 119
IVP GRANTEE(S) field of panel DSNTIPG 105
IVP PACKAGE OWNER field of panel DSNTIPG 104
IVP SQL ID field of panel DSNTIPG 104
IX_TB_PART_CONV_EXCLUDE subsystem parameter 150

J
JDBC
IBM Data Server Driver 871
JDBC AND SQLJ DLLS field of panel DSNTIPG 117
job card information fields of panel DSNTIY 350
job DSNEJBI 869
job DSNEJCB
  installing Db2 482, 483
  establishing 666
job DSNEJTM
  migrating data sharing members 720
JOB statement
  description 37

L
LANGUAGE DEFAULT field of panel DSNTIPF 197
Language Environment (LE)
  overriding run time options (migration) 512
  overriding runtime options (installation) 447
  setting up for stored procedures (installation) 447
  setting up for stored procedures (migration) 512
Language Environment run time library 920
LBACKOUT subsystem parameter 283
LC_TYPE DECP value 204
LEVEL ID UPDATE FREQ field of panel DSNTIPL1 286
library
  description 37
  distribution 44
  prefix name 42
  target 44
LIBRARY NAME PREFIX field of panel DSNTIPAP 70
LIBRARY NAME SUFFIX field of panel DSNTIPAP 70
LICENSE TERMS ACCEPTED field of panel DSNTIPO2 75
LIKE BLANK INSIGNIFICANT field of panel DSNTIP4 383
LIKE BLANK INSIGNIFICANT subsystem parameter 383
LIMIT BACKOUT field of panel DSNTIPL1 283
LINK LIST ENTRY field of panel DSNTIPM 277
LINK LIST LIBRARY field of panel DSNTIPT 117
LINK LIST SEQUENCE field of panel DSNTIPM 277
LINKNAME column
  IPLIST catalog table 926
  IPNAMES catalog table 926
  LOCATIONS catalog table 880, 895, 926
  USERSNAMES catalog table 926
links
  non-IBM Web sites 962
list structure 603
LLA (LNKLST looksaside)
  description 38
  installation 415
LMDENT option of APPL statement 891
LNKLST looksaside (LLA) 38
LNKLSTxx member of SYS1.PARMLIB
  installation 273, 415
LOAD field of panel DSNTIP6 252
LOAD LIBRARY field of panel DSNTIP7 117
load module library
  application defaults 424, 495, 717
  DSNHDECLP 424
  SDSNEXIT 38
  SDSNLINK 38
  SDSNLOAD 38
load modules
  TSO
    installation 430
    migration 489
loading
  distribution libraries 37
  target libraries 37
LOB INLINE LENGTH field of panel DSNTIPD 138
LOB_INLINE_LENGTH subsystem parameter 273
LOCAL DATE LENGTH field of panel DSNTIP4 210
local deadlock cycles 241
LOCAL TIME LENGTH field of panel DSNTIP4 210
locale
  definition 954
  specifying 954
  supporting Euros 954
LOCALE LC_CTYPE field of panel DSNTIPF 204
location name
  data sharing group 611
location names
  description 885
  updating BSDS 916
location statistics 178
LOCATIONS table
  column
    TCP/IP 926
  columns
    VTAM 895
lock
  structure
    size, determining 647
    storage requirement 647
LOCK ENTRY SIZE field of panel DSNTIP] 242
lock structure 629
  asynchronous duplexing 636
  storage estimate 649
locks
  usage, IRLM 19
LOCKS PER TABLE(SPACE) field of panel DSNTIP] 239
LOCKS PER USER field of panel DSNTIP] 240
log
  storage examples 11
LOG BACKUP STG GRP field of panel DSNTIP6 257
LOG COPY 1, BSDS 2 field 89
LOG COPY 2, BSDS 1 field 90
log mode table 892
logical unit (LU) 882
LOGMODE option of MODEENT macro 893
logon mode table entries 892

LOWER function
  ASCII
    setup 955
  Unicode
    setup 955
LRDRTHLD subsystem parameter 169
LU 6.2 communications protocols 879
LUNAME column
  LUMODES catalog table 895, 903
  MODESELECT catalog table 904
LUNAME field of panel DSNTIPR 880
LUNAME option of APPL statement
  coding values 888
  naming conventions 885
  updating BSDS 916

M
MACRO LIBRARY field of panel DSNTIP7 118
main storage 25
maintenance 6
MAINTYPE subsystem parameter 225
MANAGE REAL STORAGE field of panel DSNTIPE 168
MANAGE THREAD STORAGE field of panel DSNTIPE 169
MATERIALIZE_NODET_SQUTDF subsystem parameter 374
MAX ABEND COUNT field of panel DSNTIPX 308
MAX BATCH CONNECT field of panel DSNTIP6 166
MAX DATA CACHING field of panel DSNTIP8 225
MAX DEGREE field of panel DSNTIP8 223
MAX DEGREE FOR DPSI field of panel DSNTIP8 223
MAX IN-MEMORY SORT SIZE field of panel DSNTIPC 356
MAX KEPT DYN STMTS field of panel DSNTIPE 167
MAX OPEN CURSORS field of panel DSNTIPX 309
MAX OPEN FILE REFS field of panel DSNTIPE 167
MAX REMOTE ACTIVE field of panel DSNTIPE 165
MAX REMOTE CONNECTED field of panel DSNTIPE 165
MAX STORAGE FOR LOCKS field of panel DSNTIP9 239
MAX STORED PROC field of panel DSNTIPX 310
MAX TEMP RID field of panel DSNTIP1 161
MAX TEMP STG/AGENT field of panel DSNTIP9 160
MAX TSO CONNECT field of panel DSNTIPE 166
MAX TYPE 1 INACTIVE field of panel DSNTIP5 302
MAX USERS field of panel DSNTIPE 164
MAX UTILS PARALLELISM field of panel DSNTIP6 247
MAX_CONCURRENT_PKG_OPS subsystem parameter 374
MAXARCH subsystem parameter 291
MAXBFRU option of LINE statement 912
MAXCONQN subsystem parameter 306
MAXCONQW subsystem parameter 306
MAXCSA parameter 19
MAXDATA option of VTAM, considerations for NCP connections 914
MAXIMUM TAPE UNITS field of panel DSNTIP6 255
MAXKEEPD subsystem parameter 167
MAXPVTC option of APPL statement 891
MAXRLBK subsystem parameter 356
MAXSORT_IN_MEMORY subsystem parameter 356
MEMBER IDENTIFIER field of panel DSNTIP] 241
MEMBER NAME field of panel DSNTIPK 97
member names
  Db2 615
  MEMBER parameter of ssnmMSTR startup procedure 667
merging
  Db2 data 696
  Db2 subsystems 695
merging catalogs 695
message by identifier
DSNJ246I  728
migrating
  data sharing members  719
  job DSNJTJMA
  migrating data sharing members  719
to Db2 12  476
migrating data sharing members
CLIST  714
DSNTIJUZ  717
migrating Db2
z/OSMF  563, 570
migration
  CLIST  1
  coexistence of Db2 releases  36
data sharing group  712
    first member  713
    subsequent members  714
overview  1
  preparing  5
  release incompatibilities  481
  sample objects, required availability  481
tailing session  1
    type 1 indexes  478
  verifying  533, 733
  verifying views  537
migration jobs
  description  1
  DSNTIJEX  503
  DSNTIJIC  488
  DSNTIJIN  502
  DSNTIJMV  498
  DSNTIJPM  478
  DSNTIJTC  440, 507
  DSNTIJUZ  495
  DSNTIJVC  490
migration steps
  Job DSNTIJRT  542
    supplied routines that depend on new function  542
MINIMUM DIVIDE SCALE field of panel DSNTIP4  206
MINUTES/CHECKPOINT field of panel DSNTIPL1  282
mixed CCSIDs (coded character set identifiers)  947
MIXED DATA field of panel DSNTIPF  200
MNOTE macro error  878
MNSPAS parameter  452
mode of VTAM
  adding new modes  893
  VTAM sessions
    associating with sessions  902
    default  892
    IBMDB2LM  892
    IBMDB  892
SNASVCMG  892
mode table
  default  888, 892
  entering modes  892
MODEENT macro  892
MODENAME column
  LUMODES table  903
  MODESELECT table  904
  MODESELECT column
LUNAMES catalog table  895
MODETAB option of APPL statement  888
MODIFY IRLMPROC SET, LTE=  649
modifying OBIDs  701
MON subsystem parameter  179
MONITOR SIZE field of panel DSNTIPN  180
MONITOR TRACE field of panel DSNTIPN  179
monitoring
  VTAM buffer pools
    DISPLAY NET, BFRUSE  899
    MODIFY command of MVS  899
  multi-site updates
    APPL options  890
    MODEENT options  893
MVS
  defining Db2 to  415
MVS PARMLIB updates  273
MXDTCACH subsystem parameter  225

N
naming
  coupling facility structures  619
  IRLM  618
  member names  615
national language
  character set identifiers  947
double-byte character set identifiers  947
NCP-connected Db2 subsystems
  considerations for MAXDATA option  914
  sample definitions  914
NETID  885
NEWAUTHID
  column of USERNAMES catalog table  926
NOCACHE option of CREATE SEQUENCE statement  656
non-failure-isolated coupling facility  627
non-volatile coupling facility  632
NON-VSAM DATA field of panel DSNTIPA3  86
Notice and acceptance of OTC license panel  74
NPGRHSH subsystem parameter  374
NSYSLX parameter of IEASYSxx parmlib member  607
NUMBER OF COPIES field of panel DSNTIPH  111, 112
NUMBER OF LOCK ENTRIES field of panel DSNTIPJ  243
NUMBER OF LOGS field of panel DSNTIPL  278
NUMBER OF TCBS field of panel DSNTIPX  308
NUMLKTS subsystem parameter  239
NUMLKUS subsystem parameter  240

O
OBIDs
  modifying  701
OBJECT CREATE FORMAT field of panel DSNTIP7  140
OBJECT_CREATE_FORMAT subsystem parameter  140
OBJ REGISTRATION TABLE field of panel DSNTIPZ  348
ODBA stored procedure
  sample  785
OPERCNOS option of APPL statement  891
OPT1ROWBLOCKSORT subsystem parameter  374
OPTHINTS subsystem parameter  214
OPTIMIZATION HINTS field of panel DSNTIP8  214
OPTIMIZE EXTENT SIZING field of panel DSNTIPZ  142
OPTION, option of DSNMAPN macro  876
order-dependent transactions  660
organization application
  format command  764
  panels  797
  with IMS  764
OTC license usage panel  74
OTC_LICENSE subsystem parameter  374
OUTPUT BUFFER field of panel DSNTIPL  278
Index  979
PIU (path information unit)

description 907

relationship to MAXBRU 912

PL/I COMPILER LIBRARY field of panel DSNTIPU 130

PL/I LE PRELINKER PHASE field of panel DSNTIPU1 130

PL/I options 431

PLAN AUTH CACHE field of panel DSNTIPP 262

PLAN MANAGEMENT field of panel DSNTIP8 216

PLAN MANAGEMENT SCOPE field of panel DSNTIP8 217

PLAN option of DSNMAPN macro

description 878

installation format 876

PLAN STATEMENTS field of panel DSNTIPD 137

PLANNAME column

SYSIBM.MODESELECT catalog table 904

planning storage 8

planning XA

Global Transactions 637

xa 637

plans

invalid after migration 487

remote bind name 904

PLANS field of panel DSNTIPD 137

policy

automatic restart (ARM) 625

cfRm (coupling facility resource management) 604

DUPLEX parameter of CFRM policy 604

INITSIZE parameter of CFRM policy 604

PRELIST parameter of CFRM policy 604

SFM (Sysplex failure management) 631

SIZE parameter of CFRM policy 604

POOL THREAD TIMEOUT field of panel DSNTIP5 305

populating

cDB

connecting subsystems 895, 926

installation 469

PORT

column of LOCATIONS table 926

port numbers 922

ports

definition 918

ephemeral 918

server 918

well-known 918

PPT (z/OS program properties table) 415

PR/SM™ formulas 646

PRECOMPILER SQL LEVEL

field of panel DSNTIP41 96

prefix

active log 112

archive log 113, 288

library 42

log 113

premigration 478

PREVENT ALTER LIMITKEY field of panel DSNTIP7 150

PREVENT INDEX PART CREATE field of panel DSNTIP7 386

PREVENT ALTERTB_LIMITKEY subsystem parameter 150

PREVENT_NEW_IXCNTL_PART subsystem parameter 386

PRIMARY QUANTITY field of panel DSNTIPA 287

primary structures 633

PRIPRO1 option of MODEENT macro 893

PRIQTY subsystem parameter 287

PRIVATE_PROTOCOL subsystem parameter 374

PROC NAME field of panel DSNTIP1 235

product-sensitive programming information, described 961

PROFILE AUTOSTART field of panel DSNTIPA 187, 188
profile tables
storage requirements 19
program libraries 38
program preparation 431
programming interface information, described 961
project application
format command 764
panels 797
updating activities 802
viewing project details 802
with IMS 764
PROTECT subsystem parameter 261
provider 
web service 863
PRTCT option of APPL statement 886, 888
PSERVIC option of MODEENT macro 893
PSPI symbols 961
PTASKROL transaction type 878
PTASKROL subsystem parameter 374
PTFs 6

Q
query optimization
storage estimation 18
QUERY_ACCEL_OPTIONS subsystem parameter 227
QUERY_ACCEL_WAITFORDATA subsystem parameter 231
QUERY_ACCELERATION subsystem parameter 230, 374
QUIESC_PERIOD field of panel DSNTIPA 292
QUIESC subsystem parameter 292

R
RACF (Resource Access Control Facility)
option to specify at installation or migration 261
RANDOM ATTACH field of DSNTIPK 99
RANDOMATT subsystem parameter 99
RANDOMIZE XML DOCID field of panel DSNTIP8 218
RCT (resource control table)
installation 437
re-enabling DB2 data sharing 731
READ COPY2 ARCHIVE field of panel DSNTIPO 187
READ TAPE UNITS field of panel DSNTIPA 290
real storage 33
REAL TIME STATS field of panel DSNTIPO 219
REALSTORAGE_MANAGEMENT subsystem parameter 168
REALSTORAGE_MAX subsystem parameter 374
rebinding
avoiding in a coexistence environment 707
remote packages 954
REBUILD INDEX field of panel DSNTIP61 253
rebuild structures
SETXCF START,REBUILD command 650
specifying rebuild threshold 631
REBUILDPERCENT parameter of CFRM policy 631
REC_FASTREPLICATION subsystem parameter 250
RECALL DATABASE field of panel DSNTIPO 184
RECALL DELAY field of panel DSNTIPO 184
recording data to SMF
adding data sharing members 688
data sharing 676
RECORD MAX field of panel DSNTIPA 291
RECORDS/CHECKPOINT field of panel DSNTIPL1 282
recovering data
data sharing 731
recovery
automatic recovery of group buffer pools 628
recovery job DSNTIJDE 422
REPSHAGE subsystem parameter 225
REGISTRATION DATABASE field of panel DSNTIPZ 348
REGISTRATION OWNER field of panel DSNTIPZ 348
registration tables for DDL, recommendation 663
relational database name 885
release coexistence
Db2 36
IRLM service 710
release dependency marker 559
release incompatibilities 481
remigration
data sharing group 723
remigration 561
REMOTE COPY SW ACCEL field of panel DSNTIPL 280
REMOTE LOCATION field of panel DSNTIPE 349
removing
employee information at remote locations 819
RENAME TABLE field of panel DSNTIP2 151
RENAME TABLE subsystem parameter 151
REO (region error option)
default OPTION of DSNMAPN macro 878
installation 872
REORG DROP PBG PARTS field of panel DSNTIP63 260
REORG INDEX field of panel DSNTIP61 253
REORG LIST PROCESSING field of panel DSNTIP63 259
REORG LIST PROCESSING subsystem parameter 259
REORG MAPPING DB field of panel DSNTIP63 259
REORG PART SORT NPSI field of panel DSNTIP63 259
REORG TABLESPACE field of panel DSNTIP61 253
REORG TS NOPAD DEFAULT field of panel DSNTIP63 261
REORG.Drop_PBG_PARTS subsystem parameter 260
REORG_LIST_PROCESSING 374
REORG_MAPPING_DATABASE subsystem parameter 259
REORG_PART_SORT_NPSI subsystem parameter 259
REORG_T.S_NOPAD_DEFAULT subsystem parameter 261
REQUIRE FULL NAMES field of panel DSNTIPZ 347
Resource Access Control Facility (RACF) 261
RESOURCE AUTH NAMES field of panel DSNTIPE1 271
resource control table (RCT) 437
resource limit facility
controlling 192
default threshold for static SQL 193
resource limit facility (governor)
authorization ID 271
creating database during installation 445
specifying 193
storage estimation 18
resource management policy 630
Resource Recovery Services (RRS)
log streams used (installation) 448
log streams used (migration) 513
RESOURCE TIMEOUT field of panel DSNTIPE 233
resource translation table (RTT) 872
REST user-defined functions
creating 866
HTTP proxy support 866
SSL configuration 866
restart automatic 625
RESTART field of panel DSNTIPS 294
restart light 626
RESTORE/RECOVER field of panel DSNTIP62 254
RESTRICT_ALT_COL_FOR_DCC subsystem parameter 374
RESYNC INTERVAL field of panel DSNTIPR 300
RESYNC PORT field of panel DSNTIPR 299
sample applications (continued)

phones 808
photo application 821
viewing photos 823
printing options 798
project 802
remote access
  starting in ISPF/TSO environment 773
  resume application 821
  viewing resumes 822
  scenarios 802
SQL procedure 787
SQL procedures processor 787
stored procedure with result set 776
stored procedure without result set 774
utilities stored procedure 778
valid entries 800
verifying
  installation 733
  migration 733
  verifying installation 733
sample jobs
  DSNTIJRT 542
SAMPLE LIBRARY field of panel DSNTIPT 115
Sample trace procedures for native SQL 868
sample VTAM definitions
description 910
NCP-connected DB2 subsystems 914
samples
  DSNTESR 846
  stored procedures 773
SCA (shared communications area) 619
description 603
storage requirement 649
SCA structure 629
SDSNSBASE library 44
SDSNC.H library 44
SDSNDBRM library 44
SDSNEWS library 44
SDSNEXIT library 44, 424
SDSNINDEX library 44
SDSNINST library 44
SDSNIVPD library 44
SDSNLINK library
description 38, 44
suffix 277
SDSNLOAD library
description 44
link list options 38
SDSNSMACS library 44
SDSNSAMP library
description 44
output from panel session 358
SDSNSAMP library, naming recommendation 682
SDSNSHFL library 44
SDSNSPPF library 44

S

SAF (security authorization facility) class for coupling facility structures 607
sample application for data sharing 726
sample applications
  DATA field value 800
description 797
distributed organization 812
displaying department information 813
displaying department structure 812
DRDA access 772
field procedure 825
function keys 801
ISPF/TSO environment 759
SRCVPAC option of MODEENT macro for VTAM 983
  used to control pacing 901
SRTPool subsystem parameter 355
SSL configuration
  REST user-defined functions 866
SSM (subsystem member)
  entry in IMS.PROCLIB 872
  execution parameter 872
SSNDPAC option of MODEENT macro for VTAM 983
  recommended value 901
STAR JOIN QUERIES field of panel DSNTIP81 224
STARJOIN subsystem parameter 224
START IRLM CTRACE field of panel DSNTIP1 236
START NAMES field of panel DSNTIPS 295
start options
Db2
  adding data sharing members 691
  data sharing migration 719
  enabling data sharing 678
  installation 438
  migration 505
  VTAM 898
starting
Db2
  adding data sharing members 691
  after fallback 557
  during data sharing migration 719
  during installation 438
  during migration 505
  enabling data sharing 678
IRLM
  adding data sharing members 691
  after fallback 557
  during data sharing migration 719
  during installation 438
  during migration 505
  enabling data sharing 678
TSO
  adding data sharing members 691
  after fallback 557
  during data sharing migration 719
  during installation 438
  during migration 505
  enabling data sharing 678
startup procedures
  creating for WLM application environment (installation) 452
  creating for WLM application environment (migration) 518
STATCLGSR field of panel DSNTIP6 247
STATCLGSR subsystem parameter 247
STATFDBK_PROFILE
  subsystem parameter 220
STATFDBK_SCOPE subsystem parameter 219
STATISTICS FEEDBACK field of panel DSNTIPO 219
STATISTICS HISTORY field of panel DSNTIP6 245
STATISTICS ROLLUP field of panel DSNTIP6 245
STATISTICS SYNC field of panel DSNTIPN 179
STATISTICS TIME field of panel DSNTIPN 178
STATPGSAMP subsystem parameter 248
STD SQL LANGUAGE field of panel DSNTIP4 211
STEPLIB statement of DB2 program libraries 40
stopping
Db2
  enabling data sharing 670
stopping (continued)
Db2 (continued)
  migrating data sharing 715
  migration 493
storage
  calculating
    main 25, 32
    predefined models 10
    real 33
    virtual constraints 32
  VTAM I0BUF 907
  WLM-established address spaces 21
  working 31
  coupling facility
    calculating 639
    duplexing 633
    group buffer pool 641
    lock structure 647
    shared communications area (SCA) 649
  EDM pool
    data sharing 652
  storage message fields of panel DSNTIPC 357
  storage planning 8
  storage requirements
    calculating 9
stored procedures
  configuring Db2
    installation 445
    migration 511
  Db2 external SQL procedures Processor 509
  Db2-supplied 848
  disabling validation 848
  for XML schema support 849
  implementing RRS (installation) 448
  implementing RRS (migration) 513
  management by WLM 835
  samples 773
    callers 778
    ODBA 785
    SQL 787
    with result set 776
    without result set 774
  setting up LE (installation) 447
  setting up LE (migration) 512
  setting up WLM application environments
    WebSphere MQ user-defined functions 859
  setting up WLM application environments (installation) 452
  setting up WLM application environments (migration) 518
  use of RRSAF (installation) 448
  use of RRSAF (migration) 513
  utilities 509, 778
  Visual Explain 509
storm drain effect
  avoiding with a CICS enhancement 659
  avoiding with a CICS exit 658
  description of 658
STORTIME subsystem parameter 309
STRING DELIMITER field of panel DSNTIPF 198
structure sizes
  changing automatically 651
  changing dynamically 651
subgroup attachment name
  after disabling data sharing 728
  defining on IEFSSNxx parmlib member 654
  SUBGRP ATTACH field of panel DSNTIPK 98
SUBQ_MIDX subsystem parameter 374
SUBSYSTEM MEMBER field of panel DSNTIPM 276
SUBSYSTEM NAME field of panel DSNTIP 233
SUBSYSTEM NAME field of panel DSNTIPM 274
subsystem parameter module
data sharing members 663
installation 424
migration 495, 717
option descriptions 111
subsystem parameters
APPLCOMPAT 94
data sharing environment 663
data sharing recommendations 663
defined by DSNTIJUZ 374
DESCSTAT 207
DSMAX 351
list 396
RLFENABLE 192
RLFERRSTC 193
scope 663
STATDBK_PROFILE 220
stored procedures (installation) 446
stored procedures (migration) 511
SVOLARC 293
updating 373
updating for data sharing member 666
ZHYPERLINK 85
SUBSYSTEM SEQUENCE field of panel DSNTIPM 276
subsystems
multiple 38
name
IEFSSNxx member of SYS1.PARMLIB 415
Supplied routines
installation job 542
installing routines that depend on new function 542
SUPPRESS SOFT ERRORS field of panel DSNTIPM 277
SUPPRESS_HINT_SQLCODE_DYN subsystem parameter 374
surviving member 728
subsystem parameters
FLASHCOPY_XRCP 251
FLASHCOPY_XRCP subsystem parameter 251
SYNCLVL option of APPL statement 890
syntax diagram
how to read xiv
SYS1.PARMLIB library
updated by DSNTIJMV 415, 498
SYS1.PROCLIB library 415
SYS1.VTAMLST library 898
SYSADM 6
SYSADM authority
establish authorization IDs 429
use 469
SYSADM2 subsystem parameter 266
SYSBM.LIST table of CDB
description 926
SYSBM.USERNAMES table of CDB
description 926
SYSBM.LUMODES table of CDB
description 903
example 895, 926
SYSBM.LUMODES table of CDB
CONVLIMIT column 903
description 903
LUNAME column 903
MODENAME column 903
update 903
SYSBM.LUNAMES table of CDB
description 903
example 895
SYSBM.LUNAMES table of CDB
specifying modes 903
updating 903
SYSBM.MODESELECT table of CDB
bind plan name 904
description 904
search order 904
SYSBM.USERNAMES table of CDB
description 926
SYSLGRNX table space
increasing its size 655
SYSOPR authority
establish authorization IDs 429
SYSOPR1 subsystem parameter 266
SYSOPR2 subsystem parameter 267
Sysplex domain name, choosing for a group 611
Sysplex query parallelism
designing a database for 656
Sysplex requirements for data sharing 602
Sysplex Timer 607
SYSTEM ADMIN 1 field of panel DSNTIP1 266
SYSTEM ADMIN 2 field of panel DSNTIP1 266
system conversations
choosing default mode 895
specifying modes 903
SYSTEM LEVEL THRESHOLD field of panel DSNTIP91 163
SYSTEM MACLIB field of panel DSNTIPW 131
System Management Facility (SMF) 427
SYSTEM OPERATOR 1 field of panel DSNTIP1 266
SYSTEM OPERATOR 2 field of panel DSNTIP1 267
SYSTEM PROCEDURES field of panel DSNTIPW 132
system services address space 20
SYSTEM_LEVEL_BACKUPS subsystem parameter 254
SYSTEM-LEVEL BACKUPS field of panel DSNTIP6 254
TCP/IP
change log inventory 925
multiple stacks 934
PORT parameter 925
RESPORT parameter 925
single stack 934
termination 918
TCP/IP ALREADY VERIFIED
field of panel DSNTIP5 931
TCP/IP ALREADY VERIFIED field of panel DSNTIP5 303
TCP/IP communications
enabling 919
tcp/ip connections
defining 922
outage protection 931
overview 917
security 934
tuning 931
TCP/IP KEEPALIVE field of panel DSNTIP5 305
TCP/IP PORT field of panel DSNTIP 299
TCP/IP stacks 934
IPv4
initializing 932

T
TABLE SPACE ALLOCATION field of panel DSNTIP7 142
TABLE SPACES field of panel DSNTIPD 137
TABLES field of panel DSNTIPD 136
TABLES IN STMT field of panel DSNTIPD 138
TBSBPLOB subsystem parameter 173
TBSBPOOL subsystem parameter 171
TBSBPXML subsystem parameter 173
TCP/IP
change log inventory 925
multiple stacks 934
PORT parameter 925
RESPORT parameter 925
single stack 934
termination 918
TCP/IP ALREADY VERIFIED
field of panel DSNTIP5 931
TCP/IP ALREADY VERIFIED field of panel DSNTIP5 303
TCP/IP communications
enabling 919
tcp/ip connections
defining 922
outage protection 931
overview 917
security 934
tuning 931
TCP/IP KEEPALIVE field of panel DSNTIP5 305
TCP/IP PORT field of panel DSNTIP 299
TCP/IP stacks 934
IPv4
initializing 932

Index 985
TCP/IP stacks (continued)
IPv6
  initializing 932
VIPA
  initializing 932
TCPALVER subsystem parameter 303
TCPDPAV subsystem parameter 305
TEMP CLIST LIBRARY field of panel DSNTIPT 115
TEMP data set 358
TEMPLATE TIME field of panel DSNTIP6 246
TEMPLATE TIME subsystem parameter 246
temporary
database storage estimation 14
TEMPORARY DS UNIT NAME field of panel DSNTIP6 244
TEMPORARY UNIT NAME field of panel DSNTIPA3 85
testing SPUIF 755
thread
  space 31
TIME FORMAT field of panel DSNTIP4 209
TIME TO AUTOSTART field of panel DSNTIPI 235
TIMEOUT VALUE field of panel DSNTIPX 309
TIMESTAMP ARCHIVES field of panel DSNTIPH 113
TPN (transaction program name) 886
TPN column of LOCATIONS catalog table 895
TRACE AUTO START field of panel DSNTIPN 176
TRACE SIZE field of panel DSNTIPN 177
TRACKER SITE field of panel DSNTIPO 187
TRACSTR subsystem parameter 176
TRACTION subsystem parameter 177
transaction program name (TPN)
description 886
transmitting character data 941
TRKRSITE subsystem parameter 187
TS_COMPRESSION_TYPE field of panel DSNTIP7 143
TS_COMPRESSION_TYPE subsystem parameter 143
TSO
adding data sharing members 691
coeexistence considerations 707
enabling data sharing 678
establishing user IDs 429
installation procedures 430
data sharing 677
making load modules available 489
migration procedures 489
reconnecting 557
starting 438, 678, 691
testing 743
TSPROF option of MODEENT macro 893
TSQTY subsystem parameter 142
tuning
  VTAM
    buffer storage 899
    creating new modes 893
description 898
MODEENT macro 901
pacing count 900
SRCVPAC option of MODEENT macro 901
two-phase commit 881
DRDA (Distributed Relational Database Architecture) 881
TCP/IP 933
VTAM 890
TYPE
  option of MODEENT macro 893
type 1 indexes 478
  TYPE column
    USERNAMES catalog table 926

U
U LOCK FOR RR/RS field of panel DSNTIPI 235
UCS-2 942
Unicode
  case conversion 955
  character conversion 942
  conversion support 943
description 942
UNICODE CCSID field of panel DSNTIPF 202
UNICODE IFCIDS field of panel DSNTIPN 180
UNIT subsystem parameter 288
UNIX System Services 920, 923, 929
UNKNOWN AUTHID field of panel DSNTIPP1 271
UNREGISTERED DDL DEFAULT field of panel DSNTIPZ 347
UPDATE RATE field of panel DSNTIPL 279
updating
  activity 803
  communications database while DDF is active 906
  LUMODES table of CDB 903
  LUNAMES table of CDB 903
  MODESELECT table of CDB 904
  session limits 903
UPPER function
ASCII
  setup 955
  Unicode
    setup 955
  UR CHECK FREQ field of panel DSNTIPL1 283
  UR LOG WRITE CHECK field of panel DSNTIPL1 283
  USE DATA COMPRESSION field of panel DSNTIP71 150
  USE FOR DYNAMICRULES field of panel DSNTIP4 207
  USE PROTECTION field of panel DSNTIPP 262
  user-defined characters 947
  user-defined functions 445, 511
USERNAMES column
  IPNAMES catalog table 926
  LUNAMES catalog table 895
UTF-16 942
UTF-8 942
UTIL TEMP STORCLAS field of panel DSNTIP6 244
UTIL_DBBSG subsystem parameter 257
UTIL_LGBSG subsystem parameter 257
utilities
  migration considerations 485
  recommendations for a coexistence environment 709
  types
    CATMAINT 440, 507
    UTILITY OBJECT CONVERSION field of panel DSNTIP7 140
    UTILITY TIMEOUT field of panel DSNTIP6 246
    UTILITY_OBJECT_CONVERSION subsystem parameter 140
    UTILS_HSM_MSGDS_HLQ subsystem parameter 258
    UTIMOUT subsystem parameter 246

V
VARY DS CONTROL INTERVAL field of panel DSNTIP7 141
VARY NET command of VTAM
  ACTIVE option 898
verification jobs
DSNTEJ0 742
DSNTEJ1 743
DSNTEJIL 743, 745
DSNTEJIP 743, 746
DSNTEJ1U 747
<table>
<thead>
<tr>
<th>DSNTIEJ2A</th>
<th>748</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTIEJ2C</td>
<td>749</td>
</tr>
<tr>
<td>DSNTIEJ2D</td>
<td>749</td>
</tr>
<tr>
<td>DSNTIEJ2E</td>
<td>750</td>
</tr>
<tr>
<td>DSNTIEJ2F</td>
<td>751</td>
</tr>
<tr>
<td>DSNTIEJ2H</td>
<td>751</td>
</tr>
<tr>
<td>DSNTIEJ2P</td>
<td>752</td>
</tr>
<tr>
<td>DSNTIEJ2U</td>
<td>753</td>
</tr>
<tr>
<td>DSNTIEJ3C</td>
<td>757</td>
</tr>
<tr>
<td>DSNTIEJ3M</td>
<td>759</td>
</tr>
<tr>
<td>DSNTIEJ3P</td>
<td>757</td>
</tr>
<tr>
<td>DSNTIEJ4C</td>
<td>760</td>
</tr>
<tr>
<td>DSNTIEJ4P</td>
<td>760</td>
</tr>
<tr>
<td>DSNTIEJ5A</td>
<td>765</td>
</tr>
<tr>
<td>DSNTIEJ5C</td>
<td>765</td>
</tr>
<tr>
<td>DSNTIEJ5P</td>
<td>765</td>
</tr>
<tr>
<td>DSNTIEJ6</td>
<td>772</td>
</tr>
<tr>
<td>DSNTIEJ61</td>
<td>785</td>
</tr>
<tr>
<td>DSNTIEJ62</td>
<td>786</td>
</tr>
<tr>
<td>DSNTIEJ63</td>
<td>787</td>
</tr>
<tr>
<td>DSNTIEJ64</td>
<td>788</td>
</tr>
<tr>
<td>DSNTIEJ65</td>
<td>789</td>
</tr>
<tr>
<td>DSNTIEJ66</td>
<td>790</td>
</tr>
<tr>
<td>DSNTIEJ67</td>
<td>792</td>
</tr>
<tr>
<td>DSNTIEJ6D</td>
<td>777</td>
</tr>
<tr>
<td>DSNTIEJ6P</td>
<td>775</td>
</tr>
<tr>
<td>DSNTIEJ6R</td>
<td>780</td>
</tr>
<tr>
<td>DSNTIEJ6S</td>
<td>774</td>
</tr>
<tr>
<td>DSNTIEJ6T</td>
<td>777</td>
</tr>
<tr>
<td>DSNTIEJ6U</td>
<td>780</td>
</tr>
<tr>
<td>DSNTIEJ6V</td>
<td>782</td>
</tr>
<tr>
<td>DSNTIEJ6W</td>
<td>783</td>
</tr>
<tr>
<td>DSNTIEJ6Z</td>
<td>784</td>
</tr>
<tr>
<td>DSNTIEJ7</td>
<td>792</td>
</tr>
<tr>
<td>DSNTIEJ71</td>
<td>793</td>
</tr>
<tr>
<td>DSNTIEJ73</td>
<td>794</td>
</tr>
<tr>
<td>DSNTIEJ75</td>
<td>794</td>
</tr>
<tr>
<td>DSNTIEJ76</td>
<td>795</td>
</tr>
<tr>
<td>DSNTIEJ77</td>
<td>796</td>
</tr>
<tr>
<td>DSNTIEJ78</td>
<td>796</td>
</tr>
<tr>
<td>DSNTIEJB</td>
<td>869</td>
</tr>
<tr>
<td>DSNTESA</td>
<td>755</td>
</tr>
<tr>
<td>DSNTESC</td>
<td>755</td>
</tr>
<tr>
<td>DSNTESE</td>
<td>755</td>
</tr>
<tr>
<td>phases</td>
<td>734</td>
</tr>
<tr>
<td>programs</td>
<td>734</td>
</tr>
<tr>
<td>testing SPUFI</td>
<td>755</td>
</tr>
</tbody>
</table>

**verification testing**
- DDF 769
- DRDA 757
- dynamic SQL 757
- LOB 792
- migrating data sharing members 721

**VERIFY**
- option of APPL statement 888

**views**
- verifying 537

**VIEWS** field of panel DSNTIPD 137

**VIPA** (virtual IP address) 932

**virtual sequential access method** (VSAM) 422

**virtual storage**
- calculating constraints 32
- calculations 25
- layout 19

**Virtual Telecommunications Access Method** (VTAM) 884

**volatility of coupling facility** 632

**VPACING option** 901

**VPACING option of APPL statement** 888

**VSAM** (virtual storage access method)
- clusters 422
- CATALOG, DEFAULT AND WORK FILE DATABASE field of panel DSNTIPAD 87

**VTAM** (Virtual Telecommunications Access Method)
- APPL statement 887
  - buffer pools
    - calculating storage 907
    - increasing 899
    - monitoring 899
    - performance effect 899
  - commands
    - DISPLAY BFUSE 899
    - VARY NET 898
  - defining Db2 subsystem 887
    - log mode table, default 892
    - MODEENT macro 892
    - modes, default for DB2 892
    - sample definitions 910
- libraries
  - SYSL.SAMPLIB 892
  - SYSL.VTAMLST 898
  - naming 885, 886
  - example 887
- options
  - not applicable to Db2 892
  - parallel sessions 888
  - password 885
  - choosing 886
  - specifying in APPL statement 888
  - updating BSDS 916
  - planning considerations 884
  - session limit 906
  - calculating 906
  - mode default 888
  - start options 898
  - tracing 899
- VTAMFRR option of APPL statement 890

**W**

**WAIT FOR DATA** field of panel DSNTIP82 231

**web service consumer**
- enabling Db2 865

**web service provider**
- enabling Db2 863

**web service user-defined functions**
- enabling 860
  - enabling support for AT-TLS 862
  - enabling support for HTTPS 862
- installing 860
- supported web service calls 862

**WebSphere MQ user-defined functions**
- AMI-based 857
  - enabling 856
  - MQI-based 857
  - new versions 856
  - queue manager 859
  - updating 856
  - verifying 860
- WFDBSEP subsystem parameter 160

**WLM address spaces**
- defining 695
- verifying 695
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLM application environments</td>
<td>309</td>
</tr>
<tr>
<td>configuring for WebSphere MQ user-defined functions</td>
<td>859</td>
</tr>
<tr>
<td>defining (installation)</td>
<td>452</td>
</tr>
<tr>
<td>defining (migration)</td>
<td>518</td>
</tr>
<tr>
<td>of Db2-supplied routines</td>
<td>836</td>
</tr>
<tr>
<td>security</td>
<td>452</td>
</tr>
<tr>
<td>setting up for stored procedures (installation)</td>
<td>452</td>
</tr>
<tr>
<td>setting up for stored procedures (migration)</td>
<td>518</td>
</tr>
<tr>
<td>WLM ENVIRONMENT field of panel DSNTIPX</td>
<td>309</td>
</tr>
<tr>
<td>WLM environments setup during installation</td>
<td>457</td>
</tr>
<tr>
<td>WLM panels</td>
<td></td>
</tr>
<tr>
<td>setting up application environment (installation)</td>
<td>452</td>
</tr>
<tr>
<td>setting up application environment (migration)</td>
<td>518</td>
</tr>
<tr>
<td>WLM PROC NAME field of panel DSNTIPX</td>
<td>308</td>
</tr>
<tr>
<td>work file</td>
<td></td>
</tr>
<tr>
<td>installation job DSNTIJTM</td>
<td>441</td>
</tr>
<tr>
<td>adding data sharing members</td>
<td>693</td>
</tr>
<tr>
<td>enabling data sharing</td>
<td>680</td>
</tr>
<tr>
<td>migration job DSNTIJTM</td>
<td>509</td>
</tr>
<tr>
<td>work file database</td>
<td></td>
</tr>
<tr>
<td>connectivity requirements</td>
<td>607</td>
</tr>
<tr>
<td>considerations when disabling data sharing</td>
<td>728</td>
</tr>
<tr>
<td>considerations when re-enabling DB2 data sharing</td>
<td>731</td>
</tr>
<tr>
<td>installation job DSNTIJTM</td>
<td>441</td>
</tr>
<tr>
<td>adding data sharing members</td>
<td>693</td>
</tr>
<tr>
<td>enabling data sharing</td>
<td>680</td>
</tr>
<tr>
<td>migration job DSNTIJTM</td>
<td>509</td>
</tr>
<tr>
<td>storage estimation</td>
<td>14</td>
</tr>
<tr>
<td>WORK FILE DB field of panel DSNTIPK</td>
<td>97</td>
</tr>
<tr>
<td>workflows</td>
<td></td>
</tr>
<tr>
<td>z/OSMF samples</td>
<td>593</td>
</tr>
<tr>
<td>working storage</td>
<td>31</td>
</tr>
<tr>
<td>Workload Manager (WLM) coexistence considerations</td>
<td>704</td>
</tr>
<tr>
<td>description</td>
<td>835</td>
</tr>
<tr>
<td>management of stored procedures</td>
<td>835</td>
</tr>
<tr>
<td>service policy</td>
<td>835</td>
</tr>
<tr>
<td>WRITE TO OPER field of panel DSNTIPA</td>
<td>291</td>
</tr>
<tr>
<td>WTO ROUTE CODES field of panel DSNTIPO</td>
<td>183</td>
</tr>
<tr>
<td>WTOR ROUTE CODE field of panel DSNTIPA</td>
<td>292</td>
</tr>
</tbody>
</table>

X

X LOCK FOR SEARCHED U/D field of panel DSNTIPI  236

XCF (cross-system coupling facility) component of z/OS
Db2 group name  611
description  602

XML schema support
creating stored procedures for  849
XML_RANDOMIZE_DOCID subsystem parameter  218
XRMIOUT exit of CICS  658
XSLTRANSFORM  850

Z

z/OS
IPL  437
adding data sharing members  691
enabling data sharing  678
z/OS commands
SETSSI  682
z/OSMF
installing  564
installing or migrating  563

988  Installation and Migration Guide

z/OSMF (continued)
migrating  570

z/OSMF samples
input variable files
DSNTEVAF  593
DSNTEVIA  593
DSNTEVIVN  593
DSNTEVMFD  593
DSNTEVMN  593
DSNTEVMS  593
workflow definition files
DSNTEWIA  593
DSNTEWIN  593
DSNTEWMFD  593
DSNTEWMF  593
DSNTEWMS  593
ZHYPERLINK subsystem parameter  85
ZOSMETRICS subsystem parameter  374