

IBM Tivoli Decision Support for z/OS
Version 1.8.2

Administration Guide and Reference



IBM Tivoli Decision Support for z/OS
Version 1.8.2

Administration Guide and Reference



Before using this information and the product it supports, read the information in “Notices” on page 315.

Fifteenth Edition (March 2015)

This edition applies to version 1, release 8, modification level 2 of Tivoli Decision Support for z/OS (program number 5698-B06) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM© COPYRIGHT INTERNATIONAL BUSINESS MACHINES CORPORATION. ALL RIGHTS RESERVED. Note to U.S. Government Users—Documentation related to restricted rights—Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp. 1993, 2015

Contents

Figures	vii
--------------------------	------------

Preface	ix
--------------------------	-----------

Who should read this book	ix
Publications	ix
Tivoli Decision Support for z/OS library	ix
Accessing terminology online	x
Accessing publications online	xi
Ordering publications	xi
Accessibility	xi
Tivoli technical training	xi
Support information	xi
Conventions used in this book	xi
Typeface conventions	xii
Changes in this edition	xii

Part 1. Installing Tivoli Decision Support for z/OS 1

Chapter 1. Introducing Tivoli Decision Support for z/OS 3

Introduction to Tivoli Decision Support for z/OS	3
Introduction to Usage and Accounting Collector	4
Introduction to Tivoli Decision Support for z/OS performance features	4
Introduction to the log collector	5
Log definitions	5
Record definitions	6
Update definitions	6
Table definitions	6
Log and record procedures	6
Collect process	7
Introduction to the Tivoli Decision Support for z/OS database	9
Introduction to the administration dialog	10
Introduction to the reporting dialog	11
Introduction to the Key Performance Metrics	11
Components	12

Chapter 2. Installing Tivoli Decision Support for z/OS 15

Installation prerequisites	15
Hardware prerequisites	15
Software prerequisites	16
Step 1: Reviewing the results of the SMP/E installation	17
Tivoli Decision Support for z/OS data sets	17
Local data sets	17
Language-dependent Tivoli Decision Support for z/OS data sets	17
Step 2: Setting up security	18
Security using secondary authorization IDs	18
Security without secondary authorization IDs	20
Step 3: Initializing the DB2 database	21

Initializing DB2 database when installing Tivoli Decision Support for z/OS for the first time	22
Customizing the CICS Partitioning feature	25
Step 4: Preparing the dialog and updating the dialog profile	27
Step 5: Setting personal dialog parameters	29
Step 6: Setting up QMF	32
Step 7: Creating system tables	33
Creating and updating system tables with a batch job	35
Step 8: Customizing JCL	35
Step 9: Testing the installation of the Tivoli Decision Support for z/OS base	36
Step 10: Reviewing DB2 parameters	39
Step 11: Determining partitioning mode and keys	40
Step 12: Installing components	40
Installing the Usage and Accounting Collector	41
Customizing the Usage and Accounting Collector	42
Allocating and initializing Usage and Accounting files	44
Processing SMF data using DRLNJOB2 (DRLCDATA and DRLCACCT)	45
Running DRLNJOB3 (DRLCMONY) to create invoices and reports	47
Processing Usage and Accounting Collector Subsystems	49
Installing multiple Tivoli Decision Support for z/OS systems	49
Installing Tivoli Decision Support for z/OS features separately	50

Part 2. Installation reference 53

Chapter 3. Dialog parameters 55

Modifying the DRLFPROF dataset	55
Overview of the Dialog Parameters window	56
Dialog Parameters when QMF is used	56
Dialog Parameters when QMF is not used	57
Dialog parameters - variables and fields	58
Allocation overview	69

Chapter 4. Overview of Tivoli Decision Support for z/OS objects 71

How Tivoli Decision Support for z/OS component installation works	71
Defining definition library members with SQL	71
How Tivoli Decision Support for z/OS controls object replacement	72
Tivoli Decision Support for z/OS Version variable format	73
How Tivoli Decision Support for z/OS determines installation order	73
Defining logs with log collector language	74
Defining records with log collector language	74

Defining table spaces	75
Defining tables and updates	75
Defining updates and views	77
Defining table spaces and indexes using the GENERATE statement	77
Defining reports	78

Chapter 5. Naming convention for Tivoli Decision Support for z/OS definition members 81

Naming convention for members of DRL182.SDRLDEFS	81
Naming convention for members of DRL182.SDRLRENU	81

Part 3. Administering Tivoli Decision Support for z/OS 83

Chapter 6. Setting up operating routines 85

Collecting log data	85
Collecting data through the administration dialog	85
Using log collector language to collect data	86
The DRLJCOLL job	86
Improving collect performance	94
Administering the Tivoli Decision Support for z/OS database	95
Understanding DB2 concepts	95
Understanding how Tivoli Decision Support for z/OS uses DB2	96
Understanding table spaces	96
Calculating and monitoring table space requirements	97
Reorganizing the database	101
Backing up the Tivoli Decision Support for z/OS database	105
Recovering from database errors	107
Monitoring the size of the Tivoli Decision Support for z/OS database	108
Understanding how Tivoli Decision Support for z/OS uses DB2 locking and concurrency	109
Maintaining database security	110
Monitoring database access	110
Using available tools to work with the Tivoli Decision Support for z/OS database	111
Administering lookup and control tables	112
Administering reports	112
Running reports in batch	112
Creating report groups	120
Administering problem records	121
Reviewing exceptions and generating problem records	121
Generating problem records in batch	121

Chapter 7. Working with components 125

Installing and uninstalling a component	126
Installing a component	126
Uninstalling a component	134
Working with table space profiles	134

Reviewing Key Performance Metrics table space profiles prior to installation	135
Reviewing the GENERATE statements for table spaces, tables, and indexes	135
Working with a component definition	136
Controlling objects that you have modified	136
Viewing objects in a component	137
Viewing or editing an object definition	137
Adding an object to a component	138
Deleting an object from a component	138
Excluding an object from a component installation	139
Including an object in a component installation	139
Deleting a component	140
Creating a component	140

Chapter 8. Working with log and record definitions 143

Working with the contents of logs	144
Viewing a list of log data sets collected	144
Deleting a log data set	146
Collecting data from a log into DB2 tables	146
Displaying log statistics	148
Displaying the contents of a log	149
Creating a report on a record	150
Working with log definitions	151
Viewing and modifying a log definition	152
Working with header fields	152
Creating a log definition	153
Deleting a log definition	154
Working with record definitions in a log	154
Viewing and modifying a record definition	155
Working with fields in a record definition	157
Working with sections in a record definition	158
Creating a record definition	159
Displaying update definitions associated with a record	159
Deleting a record definition	160
Viewing and modifying a record procedure definition	160
Creating a record procedure definition	161
Deleting a record procedure definition	162

Chapter 9. Working with tables and update definitions 163

Working with data in tables	164
Displaying the contents of a table	164
Editing the contents of a table	165
Showing the size of a table	167
Recalculating the contents of a table	169
Importing the contents of an IXF file to a table	171
Exporting table data to an IXF file	172
Purging a table	172
Unloading and loading tables	172
Integration with DB2 High Performance Unload	176
Working with tables and update definitions	179
Opening a table to display columns	179
Displaying and modifying update definitions of a table	184

Displaying and editing the purge condition of a table	190
Displaying and modifying a table or index space	192
Displaying a view definition	196
Printing a list of Tivoli Decision Support for z/OS tables	197
Saving a table definition in a data set	197
Listing a subset of tables in the Tables window	198
Creating a table	198
Deleting a table or view	200
Creating a table space	200
Creating an update definition	201
Deleting an update definition	202
Administering user access to tables	202

Chapter 10. Working with the log data manager option 205

Summary of how the log data manager is used	205
Invoking the log data manager	205
Job step for recording a log data set for collection	206
Using the DRLJLDML job step	206
DRLJLDML sample job	207
Setting the parameters for job DRLJLDML.	208
Modifying log collector statements	209
Listing the data sets containing collect statements	209
Editing the collect statements	209
Adding a log ID and collect statements data set	211
Changing the collect statements data set name	211
Listing and modifying the list of log data sets to be collected	212
Listing the log data sets to be collected.	212
Modifying the log ID for a log data set.	213
Deleting information about a log data set	214
Recording a log data set to be collected again	214
Adding a log data set to be collected	214
The DRLJLDMC collect job and the parameters it uses	215
Deciding which log data sets to collect	215
Concatenation of log data sets.	215
Running collect jobs in parallel	215
DRLELDMC sample job	216
Setting the DYNAMNBR value	218
Setting the parameters for job DRLJLDMC	218
Modifying the list of successfully collected log data sets.	219
Viewing the information about successfully collected log data sets	220
Viewing the dump data set.	220
Changing the retention period of information about a log data set	220
Deleting the information about a log data set	220
Modifying the list of unsuccessfully collected log data sets	221
Viewing the unsuccessfully collected log data set	221
Viewing the dump data set.	221
Recording a log data set to be collected again	222
Deleting the information about a log data set	222

Part 4. Administration reference 223

Chapter 11. System tables and views 225

Log collector system tables	225
DRLEXPRESSIONS	225
DRLFIELDS	225
DRLLDM_COLLECTSTMT	226
DRLLDM_LOGDATASETS	227
DRLLOGDATASETS	227
DRLLOGS	228
DRLPURGECOND	229
DRLRECORDPROCS	229
DRLRECORDS	230
DRLRPROCINPUT	230
DRLSECTIONS	230
DRLUPDATECOLS	231
DRLUPDATEDISTR	231
DRLUPDATELETS	231
DRLUPDATES	232
GENERATE_PROFILES	233
GENERATE_KEYS	234
Dialog system tables	235
DRLCHARTS	235
DRLCOMPONENTS	236
DRLCOMP_OBJECTS	236
DRLCOMP_PARTS	236
DRLGROUPS	237
DRLGROUP_REPORTS	237
DRLREPORTS	237
DRLREPORT_ATTR	238
DRLREPORT_COLUMNS	239
DRLREPORT_QUERIES	239
DRLREPORT_TEXT	239
DRLREPORT_VARS	240
DRLSEARCH_ATTR	240
DRLSEARCHES	240
Views on DB2 and QMF tables	241
Views on Tivoli Decision Support for z/OS system tables	241

Chapter 12. Control tables and common tables 243

Control tables	243
DAY_OF_WEEK	243
PERIOD_PLAN	244
SCHEDULE	244
SPECIAL_DAY	245
AGGR_VALUE	245
CICS control tables	246
CICS_DICTIONARY	246
CICS_FIELD	246
Common data tables	247
Naming standard for common data tables.	247
AVAILABILITY_D, _W, _M.	247
AVAILABILITY_T	248
EXCEPTION_T	249
MIGRATION_LOG	250
Common lookup tables	250
AVAILABILITY_PARM	250
USER_GROUP	251

Chapter 13. Sample components	253
Sample component	253
SAMPLE_H, _M data tables	254
SAMPLE_USER lookup table	254
Example of table contents	254
Sample components reports	254
Sample Report 1	255
Sample Report 2	256
Sample Report 3	257

Chapter 14. Record definitions supplied with Tivoli Decision Support for z/OS.	259
SMF records	259
DFSMS/RMM records	264
IMS SLDS records	264
DCOLLECT records	267
EREP records	268
Linux on zSeries records.	268
RACF records	269
Tivoli Workload Scheduler for z/OS (OPC) records	269
VM accounting records	269
VMPRF records	270
z/VM Performance Toolkit records	270

Chapter 15. Administration dialog options and commands	273
Tivoli Decision Support for z/OS dialog options	273
Tivoli Decision Support for z/OS commands	279

Chapter 16. Administration reports	281
PRA001 - Indexspace cross-reference	281
PRA002 - Actual tablespace allocation	282
PRA003 - Table purge condition	284
PRA004 - List columns for a requested table with comments	285
PRA005 - List all tables with comments	286
PRA006 - List User Modified Objects	287

Chapter 17. Using the REXX-SQL interface	291
Calling the DRL1SQLX module	291
Input REXX variables.	292
Output REXX variables	292

Reserved REXX variable.	293
REXX example of calling DRL1SQLX	294

Chapter 18. Using the IBM DB2 Analytics Accelerator	295
Relationship of Analytics Components to non-Analytics Components	295
Configuring Analytics Components for use with IBM DB2 Analytics Accelerator	302
Collecting data for direct load to the Accelerator	304
Loading data into the Accelerator	305
Uninstalling components used with an IBM DB2 Analytics Accelerator	306

Part 5. Appendixes 307

Appendix A. Accessibility	309
Accessibility features	309
Using assistive technologies	309
IBM and accessibility	309

Appendix B. Support information	311
Searching knowledge bases	311
Searching the information center	311
Searching the Internet	311
Obtaining fixes	311
Receiving weekly support updates	312
Contacting IBM Software Support	312
Determining the business impact	313
Describing problems and gathering information	314
Submitting problems	314

Notices	315
Trademarks	317

Glossary	319
---------------------------	------------

Bibliography.	321
Tivoli Decision Support for z/OS publications	321
DB2 publications	321

Index	323
------------------------	------------

Figures

1. Tivoli Decision Support for z/OS overview	4	37. DRLJRUNS job for generating DB2 statistics	109
2. Overview of Tivoli Decision Support for z/OS data flow.	9	38. DB2I Primary Option Menu	111
3. Administration window	10	39. Converting saved graphic report data to a page segment	120
4. Introducing the Reporting dialog	11	40. Using reports in BookMaster documents	120
5. Tivoli Decision Support for z/OS Primary Menu	30	41. Using QMF to report in batch	120
6. System window - Option 1	31	42. Space pull-down	127
7. Dialog Parameters window	31	43. Installation Options window	128
8. System window - Option 3	33	44. Sample log collector messages	129
9. System Tables (not created) window	34	45. Lookup Tables window	130
10. Logs window	37	46. Editing an installation job	131
11. Sample log statistics output	37	47. Select Table window	132
12. Reports window	38	48. Tables window - showing component's lookup tables	132
13. Data Selection window.	38	49. Component window	137
14. System Tables (created) window	51	50. Data Sets window	145
15. Dialog Parameters window, when QMF is used	57	51. Collect Statistics window	145
16. Dialog Parameters window, when QMF is not used	58	52. Collect window	147
17. Tivoli Decision Support for z/OS definition member DRLISAMP, setting component definitions	72	53. Sample log statistics output	148
18. Tivoli Decision Support for z/OS definition member DRLLSAMP, defining a log type	74	54. Record Data window	149
19. Tivoli Decision Support for z/OS definition member DRLRSAMP, defining a record type	75	55. List Record window	150
20. Tivoli Decision Support for z/OS definition member DRLSSAMP, defining a table space.	75	56. Output from List record function	151
21. Tivoli Decision Support for z/OS definition member DRLTSAMP, defining tables and updates (Part 1 of 2)	76	57. Log Definition window	152
22. Tivoli Decision Support for z/OS definition member DRLTSAMP, defining tables and updates (Part 2 of 2)	77	58. Record Definitions window	155
23. Tivoli Decision Support for z/OS definition member DRLOSAMP, defining reports and report groups	79	59. Record Definition window	156
24. Tivoli Decision Support for z/OS definition member DRLQA01, report query	80	60. Field Definition window	157
25. Invoking the log collector in batch to collect data	86	61. Section Definition window	158
26. DRLJCOLL job for collecting data from an SMF data set (Part 1 of 2).	87	62. Record Procedure Definition window	161
27. DRLJCOLL job for collecting data from an SMF data set (Part 2 of 2).	88	63. Tables window	164
28. Sample collect messages	90	64. Using QMF to display a Tivoli Decision Support for z/OS table	165
29. Collect Statistics window	93	65. Editing a table in ISPF	167
30. DB2 environment for the Tivoli Decision Support for z/OS database	96	66. Table Size window.	168
31. Tablespace list window	97	67. Recalculate window	169
32. Tables window - Option 12	101	68. Condition window.	170
33. Tablespace list window	102	69. Column Values window	171
34. DRLJPURG job that uses all purge conditions	104	70. Selecting tables to unload	173
35. Tables window -Option 10	105	71. Unload Utility window	174
36. DRLJCOPY job for backing up Tivoli Decision Support for z/OS table spaces	106	72. DB2 High Performance Unload utility	178
		73. Table window	180
		74. Column Definition window.	181
		75. Add Column window	182
		76. Indexes window	183
		77. Index window	183
		78. Add Index window	184
		79. Update Definitions window.	185
		80. Update Definition window	185
		81. Abbreviations window	187
		82. Distribution window	189
		83. Apply Schedule window.	190
		84. Retention Period window	191
		85. Purge Condition window	192
		86. Tablespaces window	193
		87. Tablespace DRLxxx	193
		88. Indexes window	194
		89. Index window	195

90. Tablespace window	196	104. Retention Period window	220
91. View window	197	105. Log Data Sets Collected with Failure window	221
92. New Table window	199	106. Sample data flow	253
93. Grant Privilege window	203	107. Sample Report 1	256
94. Revoke Privilege window	204	108. Sample Report 2	257
95. Log Data Manager Main Selection window.	206	109. Sample Report 3	258
96. Collect Statements window	209	110. Part of an Indexspace Cross-reference report	282
97. Edit collect statements window	210	111. Part of an Actual Tablespace Allocation report	283
98. Add Collect Statements Definition window	211	112. Part of a Table Purge Condition report	285
99. Modify Collect Statements Definition window	212	113. Example of List columns for a requested table with comment	286
100. SMF Log Data Sets To Be Collected window	213	114. Example of List all tables with comment	287
101. Modify Log ID For a Log Data Set window.	214	115. Example of REXX-SQL interface call	294
102. Add a Data Set To Be Collected window	215		
103. Log Data Sets Collected Successfully window	219		

Preface

This book provides an introduction to IBM® Tivoli® Decision Support for z/OS® (hereafter referred to as Tivoli Decision Support for z/OS), the administration dialog, and the reporting dialog. It describes procedures for installing the base product and its features and for administering Tivoli Decision Support for z/OS through routine batch jobs and the administration dialog.

The terms listed are used interchangeably throughout the guide:

- MVS™, OS/390®, and z/OS.
- VM and z/VM®.

Who should read this book

The *Administration Guide and Reference*, SH19-6816 is for the Tivoli Decision Support for z/OS administrator, the person who initializes the Tivoli Decision Support for z/OS database and customizes and administers Tivoli Decision Support for z/OS.

Readers should be familiar with the following:

- DB2® and its utilities
- Query Management Facility (QMF™), if QMF is used with Tivoli Decision Support for z/OS
- Time Sharing Option Extensions (TSO/E)
- Restructured Extended Executor (REXX) language
- Job control language (JCL)
- Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) and its dialog manager functions

Publications

This section lists publications in the Tivoli Decision Support for z/OS library and any other related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to submit comments on Tivoli publications.

Tivoli Decision Support for z/OS library

The following documents are available in the Tivoli Decision Support for z/OS library:

- *Administration Guide and Reference*, SH19-6816.
Provides information about initializing the Tivoli Decision Support for z/OS database and customizing and administering Tivoli Decision Support for z/OS.
- *AS/400 System Performance Feature Guide and Reference*, SH19-4019.
Provides information for administrators and users about collecting and reporting performance data generated by AS/400 systems.
- *CICS Performance Feature Guide and Reference*, SH19-6820.
Provides information for administrators and users about collecting and reporting performance data generated by Customer Information and Control System (CICS®).
- *Distributed Systems Performance Feature Guide and Reference*, SH19-4018.

Tivoli Decision Support for z/OS library

Provides information for administrators and users about collecting and reporting performance data generated by operating systems and applications running on a workstation.

- *Guide to Reporting*, SH19-6842.

Provides information for users who display existing reports, for users who create and modify reports, and for administrators who control reporting dialog default functions and capabilities.

- *IMS Performance Feature Guide and Reference*, SH19-6825.

Provides information for administrators and users about collecting and reporting performance data generated by Information Management System (IMS).

- *Language Guide and Reference*, SH19-6817.

Provides information for administrators, performance analysts, and programmers who are responsible for maintaining system log data and reports.

- *Messages and Problem Determination*, SH19-6902.

Provides information to help operators and system programmers understand, interpret, and respond to Tivoli Decision Support for z/OS messages and codes.

- *Network Performance Feature Installation and Administration*, SH19-6901.

Provides information for network analysts or programmers who are responsible for setting up the network reporting environment.

- *Network Performance Feature Reference*, SH19-6822.

Provides reference information for network analysts or programmers who use the Network Performance feature.

- *Network Performance Feature Reports*, SH19-6821.

Provides information for network analysts or programmers who use the Network Performance feature reports.

- *Resource Accounting for z/OS*, SH19-4495

Provides information for users who want to use Tivoli Decision Support for z/OS to collect and report performance data generated by Resource Accounting.

- *Resource Accounting*, SH19-6818

Provides information for performance analysts and system programmers who are responsible for meeting the service-level objectives established in your organization.

- *System Performance Feature Guide*, SH19-6819.

Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for z/OS to analyze z/OS, z/VM, zLinux, and their subsystems, performance data.

- *System Performance Feature Reference Volume I*, SH19-4494

Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for z/OS to analyze z/OS, z/VM, zLinux, and their subsystems, performance data.

- *System Performance Feature Reference Volume II*, SC23-7966.

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

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

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:

<http://www.ibm.com/software/tivoli/library/>

Ordering publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.com/publications/servlet/pbi.wss>

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the Accessibility Appendix in the *Administration Guide and Reference*.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

<http://www.ibm.com/software/tivoli/education/>

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see Appendix B, “Support information,” on page 311.

Conventions used in this book

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

The following terms are used interchangeably throughout this book:

- MVS, OS/390, and z/OS.
- VM and z/VM.

Conventions used in this book

Except for editorial changes, updates to this edition are marked with a vertical bar to the left of the change.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip**, and **Operating system considerations**)
- Column headings in a table
- Keywords and parameters in text

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Changes in this edition

This edition is an update of the previous edition of the same book. New additions are:

Part 1. Installing Tivoli Decision Support for z/OS

“Introduction to the Key Performance Metrics Components” on page 12

“Step 11: Determining partitioning mode and keys” on page 40

Part 2. Installation Reference

“Defining table spaces and indexes using the GENERATE statement” on page 77

Part 4. Administering Tivoli Decision Support for z/OS

“Working with table space profiles” on page 134

“Reviewing Key Performance Metrics table space profiles prior to installation” on page 135

“Reviewing the GENERATE statements for table spaces, tables, and indexes” on page 135

Part 5. Installation reference

“GENERATE_PROFILES” on page 233

`"GENERATE_KEYS"` on page 234

Typeface conventions

Part 1. Installing Tivoli Decision Support for z/OS

Chapter 1. Introducing Tivoli Decision Support for z/OS

IBM Tivoli Decision Support for z/OS (hereafter referred to as Tivoli Decision Support for z/OS) enables you to effectively manage the performance of your system by collecting performance data in a DB2 database and presenting the data in a variety of formats for use in systems management. After reading this topic, you should have a basic understanding of Tivoli Decision Support for z/OS and be ready to install it.

This topic describes:

- How Tivoli Decision Support for z/OS works.
- Introduction to the Usage and Accounting Collector.
- Tivoli Decision Support for z/OS features.
- The log collector
- The Tivoli Decision Support for z/OS database.
- The administration dialog.
- The reporting dialog.

Introduction to Tivoli Decision Support for z/OS

Tivoli Decision Support for z/OS has two basic functions:

1. Collecting systems management data into a DB2 database.
2. Reporting on the data.

Tivoli Decision Support for z/OS consists of a base product and several optional features.

The Tivoli Decision Support for z/OS base can generate graphic¹ and tabular reports using systems management data it stores in its DB2 database. The base product includes the administration dialog, the reporting dialog, and the log collector, all of which interact with a standard DB2 database.

Figure 1 on page 4 shows an overview of Tivoli Decision Support for z/OS

1. To generate and display graphic reports, Tivoli Decision Support for z/OS uses Graphical Data Display Manager (GDDM). If you are using Tivoli Decision Support for z/OS without QMF, GDDM is not required. If GDDM is not used, all reports are displayed in tabular form.

Introduction to Tivoli Decision Support for z/OS

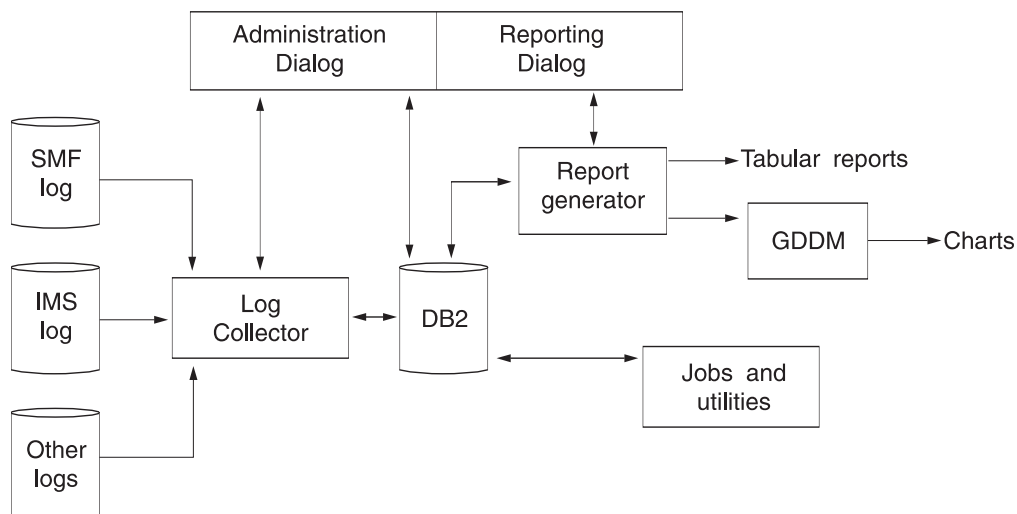


Figure 1. Tivoli Decision Support for z/OS overview

Tivoli Decision Support for z/OS (from version 1.8) supports large format input and output sequential data sets (greater than 65,535 tracks or 4369 cylinders per volume).

Introduction to Usage and Accounting Collector

The CIMS Lab Mainframe collector is incorporated into Tivoli Decision Support and called the Usage and Accounting Collector. This extracts z/OS accounting data which is used to populate Tivoli Usage and Accounting Manager databases on distributed platforms. The Usage and Accounting Collector does not require DB2 as prerequisite software on z/OS.

For a description of the Usage and Accounting Collector, see “System Overview” in the *Usage and Accounting Collector User Guide*.

For information on how to install the Usage and Accounting Collector, see “Installing the Usage and Accounting Collector” on page 41.

Note: Spectrum Writer is not included with UAC. Former CIMS Lab customers have a perpetual license for Spectrum Writer and should retain the CIMS Lab data sets so that they can make use of it. For support of Spectrum Writer, contact Pacific Systems. Customers that require access to CIMS Mainframe 12.2.1 should contact IBM support.

Introduction to Tivoli Decision Support for z/OS performance features

Tivoli Decision Support for z/OS performance features provide DB2 table definitions and table update instructions for collecting required systems management data. They also provide predefined queries, forms, and reports for presenting that data.

Resource Accounting for z/OS is part of the Tivoli Decision Support for z/OS base function.

The following performance features are additional to the base function:

- AS/400 System Performance feature.
- Customer Information Control System (CICS) Performance feature.

Introduction to Tivoli Decision Support for z/OS performance features

- Distributed Systems Performance feature.
- Information Management System (IMS) Performance feature.
- Network Performance feature.
- System Performance feature.

These features are used to collect and report on systems management data, such as System Management Facility (SMF) data or IMS log data.

Each Tivoli Decision Support for z/OS performance feature has *components*, which are groups of related Tivoli Decision Support for z/OS definitions. For example, the z/OS Performance Management (MVSPM) component consists of everything Tivoli Decision Support for z/OS needs to collect log data and create reports showing z/OS performance characteristics.

Introduction to the log collector

At the center of Tivoli Decision Support for z/OS is the *log collector* program that reads and processes performance data. Log collector tasks are controlled by log, record, update, and other definitions in Tivoli Decision Support for z/OS system tables. For more information, see “Log collector system tables” on page 225. You can add or modify definitions with both the administration dialog (see “Introduction to the reporting dialog” on page 11) and *log collector language* statements. For information on the administration dialog, see “Introduction to the administration dialog” on page 10.

Tivoli Decision Support for z/OS provides both batch and interactive processing of log collector language statements. For a description of the log collector and the language, refer to the *Language Guide and Reference*.

The key function of the log collector is to read data and store it in *data tables* in the Tivoli Decision Support for z/OS database. The log collector groups the data by hour, day, week, or month. It computes sums, maximum or minimum values, averages, and percentiles, and calculates resource availability. The *collect process*, also referred to as *collecting data* or as *collect*, includes gathering, processing, and storing the data.

Log definitions

Tivoli Decision Support for z/OS gathers performance data about systems from sequential data sets such as those written by SMF under z/OS, or by the Information Management System (IMS). These data sets are called *log data sets* or *logs*.

To collect log data, Tivoli Decision Support for z/OS needs log descriptions. The log collector stores descriptions of logs as *log definitions* in the Tivoli Decision Support for z/OS database. All log definitions used by Tivoli Decision Support for z/OS features are provided with the base product.

The administration dialog enables you to create log definitions or modify existing ones. For more information, see Chapter 8, “Working with log and record definitions,” on page 143.

The log collector language statement, DEFINE LOG, also enables you to define logs. For more information, refer to the description of defining logs in the *Language Guide and Reference*.

Record definitions

Each record in a log belongs to one unique record type. Examples of record types include SMF record type 30, generated by z/OS, and SMF record type 110, generated by CICS. For Tivoli Decision Support for z/OS to process a record, the record type must be defined. Detailed record layouts, field formats, and offsets within a record, are described in *Tivoli Decision Support for z/OS record definitions*. All record definitions used by Tivoli Decision Support for z/OS features are provided with the base product.

The administration dialog enables you to create and modify record definitions. For more information, see Chapter 8, “Working with log and record definitions,” on page 143.

The log collector language statement, `DEFINE RECORD`, also enables you to define records. For more information, refer to the description of defining records in the *Language Guide and Reference*.

Update definitions

Instructions for processing data and inserting it into tables in the Tivoli Decision Support for z/OS database are provided in *update definitions*. Each update definition describes how data from a source (either a specific record type, or a row of a table) is manipulated and inserted into a target (a row in a table). The update definitions used by a Tivoli Decision Support for z/OS component are provided with the feature that contains the component.

The administration dialog enables you to create update definitions or modify them. For more information, see “Displaying and modifying update definitions of a table” on page 184.

The log collector language statement, `DEFINE UPDATE`, also enables you to define updates. For more information, refer to the description of defining updates in the *Language Guide and Reference*.

Table definitions

Tivoli Decision Support for z/OS stores data collected from log data sets in its database tables. It also stores Tivoli Decision Support for z/OS system data in system tables and site-specific operating definitions in lookup and control tables. A *table definition* identifies the database and table space in which a table resides, and identifies columns in the table. The table definitions used exclusively by the feature components in Tivoli Decision Support for z/OS are provided with the feature.

The administration dialog enables you to create or modify lookup and data table definitions. For more information, see Chapter 9, “Working with tables and update definitions,” on page 163.

Log and record procedures

Log procedures and *record procedures* are user exit programs for specific data collection scenarios. Record procedures work on specific record types. Log procedures work on an entire log. The log and record procedures used by Tivoli Decision Support for z/OS features are provided with the base product.

For information about creating log and record procedure exits, refer to the *Language Guide and Reference*.

The administration dialog enables you to view and modify record procedure definitions, to identify record definitions that require processing by record procedures, and to define record definitions that are output from a record procedure. For more information, see “Viewing and modifying a record procedure definition” on page 160.

Collect process

When definitions exist for a log, the log records, the log update instructions for record data, and target data tables, you can collect data from that log. You start the collect process:

- From the administration dialog.
- With the log collector language statement COLLECT.

The log collector retrieves stored definitions and performs the data collection that they define.

Figure 2 on page 9 shows the collect process. Tivoli Decision Support for z/OS processes data in these steps:

1. The operating system or other program writes data to a sequential log data set, which is the input to Tivoli Decision Support for z/OS.
2. You initiate the collect either through the dialog or by using a Tivoli Decision Support for z/OS language statement in a job, identifying a specific log type definition.
3. Optionally, the log definition might process the log data with a user exit program; a log procedure. If the log definition calls a log procedure:
 - a. The log procedure receives each record in the log as input.
 - b. Output from a log procedure varies in format and is usually a record mapped by a Tivoli Decision Support for z/OS record definition.
4. Tivoli Decision Support for z/OS looks for record definitions associated with the log definition in its system tables. It applies those record definitions to specific record types from the log or log procedure.
5. Optionally, a record definition might require processing by a user exit program; a record procedure. If a record definition requires processing by a record procedure:
 - a. The record procedure receives only a specific record type and is not called for other record types.
 - b. Output from a record procedure varies in format and is usually a record mapped by a Tivoli Decision Support for z/OS record definition.
6. Tivoli Decision Support for z/OS applies a specific update definition to each known record type and performs the data manipulations and database updates as specified.
7. Tivoli Decision Support for z/OS often selects data from lookup tables to fulfill the data manipulations that update definitions require.
8. Tivoli Decision Support for z/OS writes non-summarized and first-level summarized data to data tables specified by the update definitions.
9. Tivoli Decision Support for z/OS uses updated tables as input for updating other, similar tables that are for higher summary levels. If update definitions specify data summarization:
 - a. Tivoli Decision Support for z/OS selects data from a table as required by the update definitions and performs required data summarization.
 - b. Tivoli Decision Support for z/OS updates other data tables as required by update definitions.

Introduction to the log collector

(Tivoli Decision Support for z/OS might select data from lookup tables during this process, but this step is not shown in Figure 2 on page 9.)

10. After Tivoli Decision Support for z/OS stores the data from a collect, you can display reports on the data. Tivoli Decision Support for z/OS uses a query to select the data for the report.
11. Optionally, Tivoli Decision Support for z/OS might select data from lookup tables specified in the query.
12. Tivoli Decision Support for z/OS creates report data, displaying, printing, and saving it as you requested.

For more information about collecting log data, see Chapter 6, “Setting up operating routines,” on page 85.

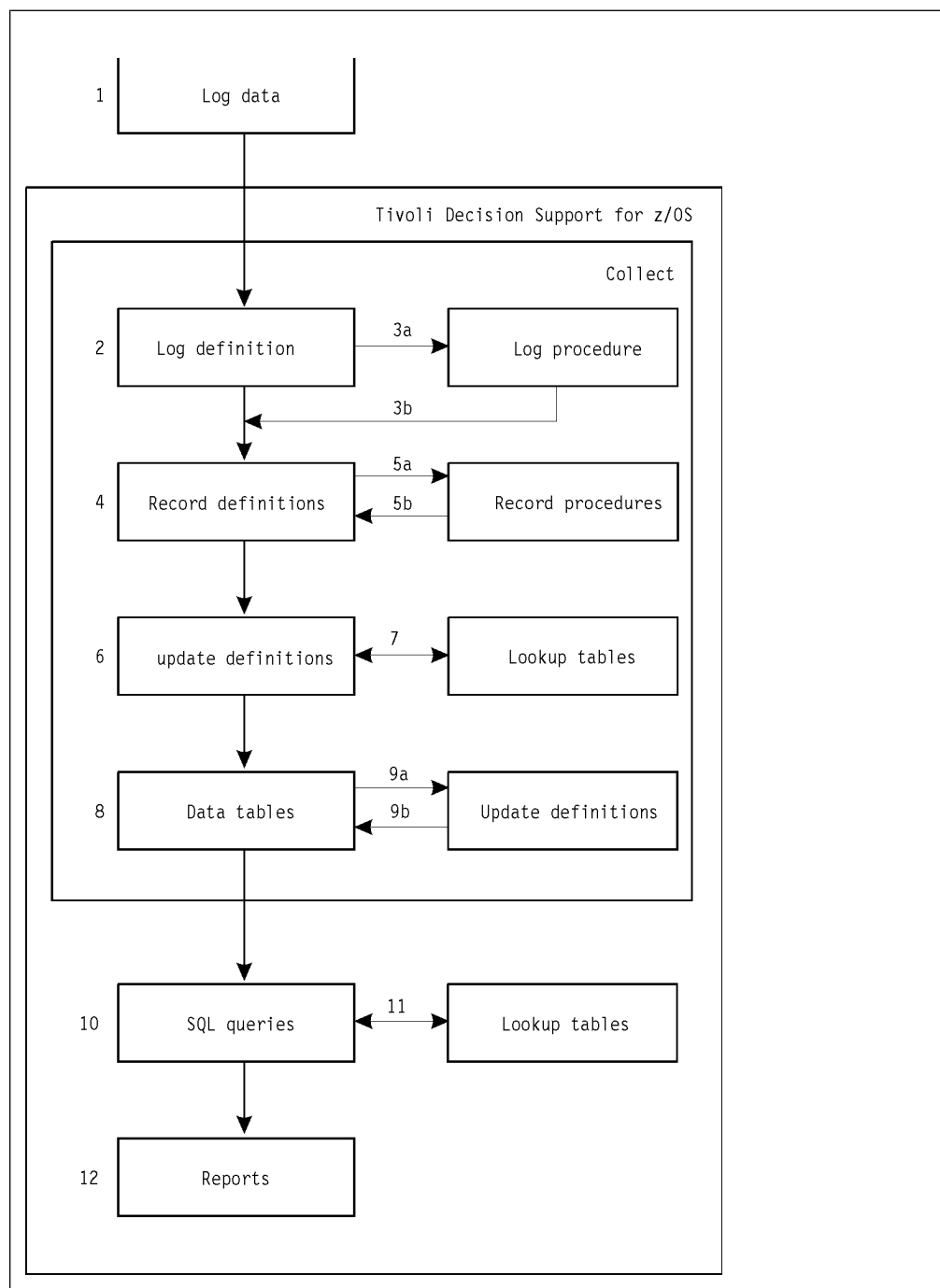


Figure 2. Overview of Tivoli Decision Support for z/OS data flow

Introduction to the Tivoli Decision Support for z/OS database

The IBM Tivoli Decision Support for z/OS database contains system tables, lookup tables, and collected data. Log collector processing transforms large amounts of log data into useful information about your systems and networks. The volume of this information in the data tables is less than the volume of data read from logs.

Tivoli Decision Support for z/OS stores data that it collects in hourly, daily, weekly, and monthly tables, and in non-summarized tables. It maintains groups of tables that have identical definitions except for their summarization levels. For example,

Introduction to the Tivoli Decision Support for z/OS database

the EREP component of the System Performance feature creates the data tables EREP_DASD_D and EREP_DASD_M, which differ only because one contains daily data and the other, monthly data.

Because the Tivoli Decision Support for z/OS database is relational, you can:

- Combine information from any of your systems into a single report.
- Summarize by system within department, by department within system, or by whatever grouping is required.

You can keep data tables containing historical data for many years without using much space. The database size depends mainly on the number of short-term details you keep in it and not on summarized weekly or monthly data.

The Tivoli Decision Support for z/OS database contains operating definitions in its system tables. These definitions include those for logs, records, updates, and tables shipped with Tivoli Decision Support for z/OS. The database also contains lookup tables of parameters that you supply, such as performance objectives or department and workload definitions for your site.

Introduction to the administration dialog

The administration dialog enables you to carry out the following tasks:

1. Install and customize Tivoli Decision Support for z/OS and its features.
2. Install and customize Tivoli Decision Support for z/OS components.
3. Work with log and record definitions.
4. Work with tables in the Tivoli Decision Support for z/OS database.
5. Create and run reports.

All of these options are available from the Administration window (Figure 3).

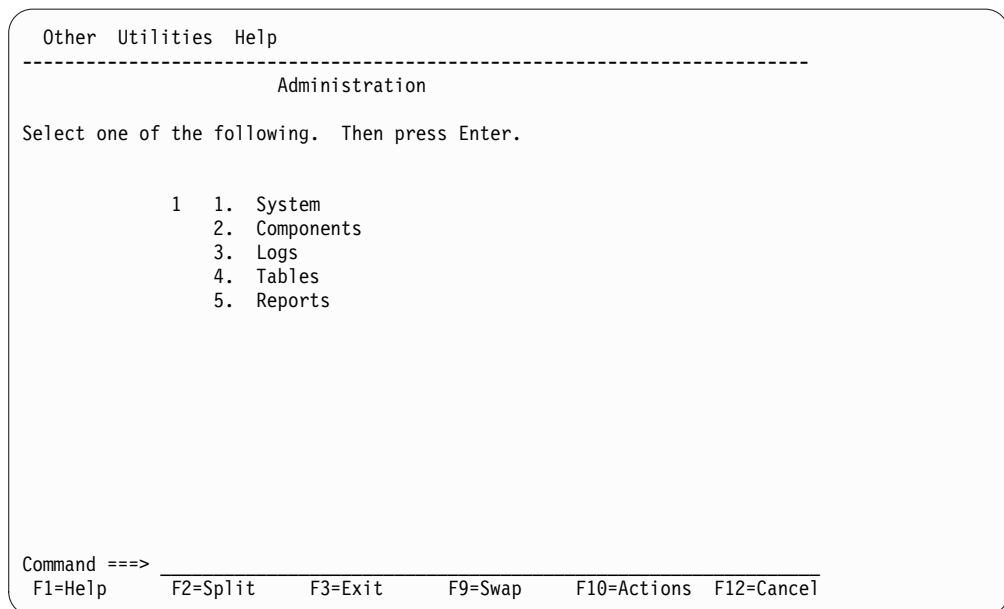


Figure 3. Administration window

Introduction to the reporting dialog

The Tivoli Decision Support for z/OS reporting dialog enables you to display reports that present the log data stored in the product database. When you use the reporting dialog to display or print a report, Tivoli Decision Support for z/OS runs a *query* associated with the report to retrieve data from the database, and then displays, or prints, the results according to an associated *form*. If your installation uses QMF with Tivoli Decision Support for z/OS, QMF is started up when you work with queries and reports. Otherwise, Tivoli Decision Support for z/OS uses its own report generator.

Figure 4 shows the Reporting dialog.

```

Options  Help
-----
                                Reporting Dialog Defaults

Type information.  Then press Enter to save defaults.

Entry to dialog . . . 1  1.  Display of previous selection
                        2.  Display of all reports
                        3.  Display of a selected group of reports

Group ID . . . . . _____ + (required if group selected)
Group owner . . . . . _____ (blank for public group)

Display of this window 1  1.  No display
                        2.  Display at exit from dialog
                        3.  Display at entry to dialog

Confirmation of exit  1  1.  Yes
                        2.  No

F1=Help    F2=Split    F4=Prompt    F9=Swap    F12=Cancel

Command ==>>>
F1=Help    F2=Split    F3=Exit     F9=Swap    F10=Actions F12=Cancel

```

Figure 4. Introducing the Reporting dialog

When you produce a report, you can specify values for the query that is used to select specific rows of data. You can display, print, or save, the retrieved data in either a tabular or a graphic² report format.

A report can consist of these items, which are identified in the *report definition*:

- A query for selecting data (required).
- A form that formats the data and specifies report headings and totals.
- Graphical Data Display Manager (GDDM) format for a graphic report.
- Report attributes (for creating logical groups of reports).
- Report groups to which the report belongs.
- Variables in the report.

2. To generate and display graphic reports, Tivoli Decision Support for z/OS uses Graphical Data Display Manager (GDDM). If you are using Tivoli Decision Support for z/OS without QMF, GDDM is not required. If GDDM is not used, all reports are displayed in tabular form.

Introduction to the reporting dialog

When installing a component, you install a comprehensive set of predefined report queries, forms, and, optionally, GDDM formats for the component. The reporting dialog enables you to:

- Define new report definitions or modify existing ones.
- Define new queries and forms or modify existing ones, using QMF or the Tivoli Decision Support for z/OS built-in report generator.
- Display reports.
- Define reports for batch execution.

The *Guide to Reporting*, SH19-6842 describes the host reporting dialog. For a description of using the Common User Access (CUA) interface presented in Tivoli Decision Support for z/OS windows and helps, refer to the "Getting Started" section of that book.

Introduction to the Key Performance Metrics Components

Tivoli Decision Support for z/OS Version 1.8.2 introduces four new components, called the Key Performance Metrics (also referred to as KPM) components. Specifically, there is one KPM component for each of z/OS, DB2, CICS, and IMS. Within the Tivoli Decision Support for z/OS component list, you will see the new components named as follows:

- Key Performance Metrics – z/OS
- Key Performance Metrics – CICS
- Key Performance Metrics – DB2
- Key Performance Metrics – IMS

These components are designed to only collect data that is considered to be key metrics for the monitoring of these subsystems. They can be installed stand alone, or they can be installed along with the corresponding existing base component. For example, the DB2 KPM component could be installed with or without the existing DB2 component being installed. Note that if you had both the DB2 KPM component and the existing base DB2 component installed, at collect time you only need to collect the SMF log the once to populate the data tables for both components.

The number of tables and columns within each of these tables will be significantly reduced in each of the KPM components. For this reason, the performance of collecting data into these components should be significantly improved when compared against their associated existing base components. For users who only reference metrics from the KPM tables, collecting only the KPM components should result in considerable CPU and elapsed time savings at collect time when compared to collecting the corresponding base components.

For details on each of the individual KPM components, refer to the appropriate guide in the table below.

Table 1. KPM components

KPM Component	Guide
Key Performance Metrics – z/OS	System Performance Feature Reference Volume I
Key Performance Metrics – CICS	CICS Performance Feature Guide and Reference
Key Performance Metrics – DB2	System Performance Feature Reference Volume I
Key Performance Metrics - IMS	IMS Performance Feature Guide and Reference

| Each KPM component uses table space profiles which allow the table, table space,
| and index settings within each KPM component to be easily modified in one place.
| Before installing the KPM components, refer to the topic “Working with table space
| profiles” on page 134.

Chapter 2. Installing Tivoli Decision Support for z/OS

This section describes how to install Tivoli Decision Support for z/OS for the first time.

If you are migrating to Tivoli Decision Support for z/OS Version 1.8.2, do not follow the installation instructions in this chapter, but follow the migration instructions documented in tech note <http://www.ibm.com/support/docview.wss?uid=swg21699114>.

The initial installation process starts *after* a system programmer has performed the SMP/E installation. The SMP/E installation of the Tivoli Decision Support for z/OS base and its features is described in the *Tivoli Decision Support for z/OS Program Directory*. The installation prerequisites from the *Tivoli Decision Support for z/OS Program Directory* are summarized in this section.

This section describes the following installation tasks:

- “Step 1: Reviewing the results of the SMP/E installation” on page 17
- “Step 2: Setting up security” on page 18
- “Step 3: Initializing the DB2 database” on page 21
- “Step 4: Preparing the dialog and updating the dialog profile” on page 27
- “Step 5: Setting personal dialog parameters” on page 29
- “Step 6: Setting up QMF” on page 32
- “Step 7: Creating system tables” on page 33
- “Step 8: Customizing JCL” on page 35
- “Step 9: Testing the installation of the Tivoli Decision Support for z/OS base” on page 36
- “Step 10: Reviewing DB2 parameters” on page 39
- “Step 11: Determining partitioning mode and keys” on page 40
- “Step 12: Installing components” on page 40
- “Installing the Usage and Accounting Collector” on page 41
- “Installing multiple Tivoli Decision Support for z/OS systems” on page 49
- “Installing Tivoli Decision Support for z/OS features separately” on page 50

You can also use this information to install other Tivoli Decision Support for z/OS systems or to install features that you did not install with the Tivoli Decision Support for z/OS base. For example; how to install the Usage and Accounting Collector.

Installation prerequisites

This section lists the hardware and software prerequisites.

Hardware prerequisites

Tivoli Decision Support for z/OS can run in any hardware environment that supports the required software.

Installing Tivoli Decision Support for z/OS

Software prerequisites

From Tivoli Decision Support for z/OS Version 1.8, the Usage and Accounting Collector (formerly CIMS mainframe) has been included in the base feature of the product. The Usage and Accounting Collector has different software prerequisites to the original or "classic" version of Tivoli Decision Support for z/OS.

The minimum requisites for Tivoli Decision Support for z/OS (excluding Usage and Accounting Collector) to install successfully are:

Program number	Product name and minimum VRM/service level
5625 – DB2	IBM DB2 Universal Database™ for z/OS Version 10
5615-DB2	IBM DB2 for z/OS Version 11
5694-A01	z/OS Version 1.13
5650-ZOS	z/OS Version 2.1

The functional requisites that Tivoli Decision Support for z/OS needs at run time for its specific functions to work are:

Product number	Product name and minimum VRM/service level	Function
5625-DB2	Query Management Facility (QMF) for z/OS Version 8	Generate and view reports
5695-167	Graphical Data Display Manager (GDDM) Version 3.2	Display reports in graphical format
5668-812	GDDM – PGF Version 2.1.3	Transform reports into graphical format
5698-SD9	Tivoli Information Management for z/OS Version 7.1	Generate problem reports from Tivoli Decision Support for z/OS data
5722-SS1	OS/400® Version 5.1	AS/400 system performance
5685-108	NetView® FTP Version 2.1	AS/400 system performance
5733-196	NetView FTP/400 Version 3	AS/400 system performance
5655-AA1	DB2 High Performance Unload (HPU) Version 4.2	Unload DB2 data enhancement
Any one of the following:		
5765-E61	AIX 5L™ Version 5.1	Distributed Systems
	HP – UX** Version 11-i	Distributed Systems
	Sun Solaris Version 9	Distributed Systems
	Linux RedHat Version 7.1 (Kernel 2.4.2)	Distributed Systems
	Linux SUSE Version 7.1 (Kernel 2.4.0)	Distributed Systems
	SLES 8 for zSeries	zLinux Systems
	RedHat Enterprise Linux 3 for zSeries	zLinux Systems

The minimum requisites for the Usage and Accounting Collector to install successfully are:

Program number	Product name and minimum VRM/service level
5694-A01	z/OS Version 1.13

Step 1: Reviewing the results of the SMP/E installation

About this task

The following default data set names are created during SMP/E installation of the Tivoli Decision Support for z/OS base and its features:

Tivoli Decision Support for z/OS data sets

Data set name	Description
DRL182.SDRLCNTL	Sample jobs and DB2 DBRM module
DRL182.SDRLDEFS	Definitions of records, tables, and other objects
DRL182.SDRLEXEC	REXX execs
DRL182.SDRLLOAD	Load modules
DRL182.SDRLSKEL	ISPF skeletons
DRL182.SDRLA400	OS/400
DRL182.SDRLWS	Workstation

Local data sets

Data set name	Description
&HLQ.LOCAL.ADMCFORM	Local GDDM-Presentation Graphics Facility (GDDM-PGF) interactive chart utility (GDDM/ICU) formats
&HLQ.LOCAL.CHARTS	Saved graphic reports (GDDM ADMGDF format)
&HLQ.LOCAL.CNTL	Local Tivoli Decision Support for z/OS jobs
&HLQ.LOCAL.DEFS	Local Tivoli Decision Support for z/OS definitions
&HLQ.LOCAL.EXEC	Local Tivoli Decision Support for z/OS execs
&HLQ.LOCAL.MESSAGES	Messages sent through the dialog
&HLQ.LOCAL.REPORTS	Saved tabular reports
&HLQ.LOCAL.USER.DEFS	Local Tivoli Decision Support for z/OS user/alter definitions

Language-dependent Tivoli Decision Support for z/OS data sets

The last three letters in these data set names indicate the language version. *xxx* is ENU for English and JPN for Japanese. For example, SDRLRENU contains the English report definition files. The corresponding Japanese version is SDRLRJPN.

Data set name	Description
---------------	-------------

Installing Tivoli Decision Support for z/OS

DRL182.SDRLFxxx
GDDM/ICU formats
DRL182.SDRLMxxx
ISPF messages
DRL182.SDRLPxxx
ISPF windows
DRL182.SDRLRxxx
Definitions of reports
DRL182.SDRLTxxx
ISPF tables

Step 2: Setting up security

About this task

This topic describes how you can protect Tivoli Decision Support for z/OS data sets and the database.

Use RACF® or a similar product to protect the Tivoli Decision Support for z/OS data sets. Administrators and users must have read access to the DRL182 data sets and update access to the local data sets.

The data in the database is protected by DB2. Administrators and users must be granted DB2 privileges to be able to access the data, as follows:

- Administrators need SYSADM (system DB2 administrator authority for the Tivoli Decision Support for z/OS database. They also need the ability to use the prefixes of Tivoli Decision Support for z/OS tables (DRLSYS and DRL) as authorization IDs in DB2.
- Users need read access to the tables they use to produce reports, and update access to some of the Tivoli Decision Support for z/OS system tables (to be able to create their own reports).
- The user IDs that you use for Tivoli Decision Support for z/OS production jobs, such as collect, need DBADM authority.

This step describes two ways you can define authorities for Tivoli Decision Support for z/OS administrators and users:

- Using secondary authorization IDs.
- Without secondary authorization IDs.

Find out through the DB2 system administrator whether secondary authorization IDs are used on your DB2 system.

Note: If you are defining authorities without using secondary user IDs, the installation process is slightly different. See “Security without secondary authorization IDs” on page 20 for more information.

Security using secondary authorization IDs

About this task

The most efficient way to give users privileges is to use secondary authorization IDs in DB2. With this method, privileges are granted to group IDs rather than user IDs, and all users who can use these secondary authorization IDs get the privileges.

The secondary authorization IDs a user has access to can be controlled in different ways. If you have RACF installed, users can usually use the RACF groups that they are connected to as secondary authorization IDs. If RACF is not installed, secondary authorization IDs can be assigned by the DB2 authorization exit.

This topic describes how to define the secondary authorization IDs using RACF. If you assign secondary authorization IDs in another way, consult your DB2 system administrator.

Procedure

1. Create three RACF groups. The default RACF group IDs are DRL, DRLSYS, and DRLUSER

```
ADDGROUP DRL DATA ('Tivoli Decision Support for z/OS TABLES')
ADDGROUP DRLSYS DATA ('Tivoli Decision Support for z/OS
                      SYSTEM TABLES')
ADDGROUP DRLUSER DATA ('Tivoli Decision Support for z/OS USERS')
```

The IDs DRL and DRLSYS are also prefixes for the Tivoli Decision Support for z/OS DB2 tables. If you plan to change the prefixes for Tivoli Decision Support for z/OS system tables and views (DRLSYS) or for other Tivoli Decision Support for z/OS tables and views (DRL) in “Step 3: Initializing the DB2 database” on page 21, use your values as RACF group IDs.

If all users on your system need access to the Tivoli Decision Support for z/OS data, you do not need the DRLUSER group. If different users need access to different sets of tables, you can define several RACF group IDs, such as DRLMVS and DRLCICS, instead of the DRLUSER group. You can use either RACF commands or RACF dialogs to specify security controls. These commands are samples. You may have to specify additional operands to comply with the standards of your organization.

2. Connect Tivoli Decision Support for z/OS administrators to all three groups. Use RACF commands or RACF dialogs to connect user IDs to a group. This command is a sample.

```
CONNECT (user_ID1 user_ID2 ...) GROUP(DRLUSER)
```

Note: VIEWER users need to be connected to the above three groups (DRL , DRLSYS , DRLUSER).

3. Connect Tivoli Decision Support for z/OS (not VIEWER) users to the DRLUSER group only.

Use RACF commands or RACF dialogs to connect user IDs to a group. This command is a sample.

```
CONNECT (user_ID1 user_ID2 ...) GROUP(DRLUSER)
```

4. If you use different RACF group IDs, be sure to use them throughout all the steps listed .
5. If you use other group IDs than DRLUSER, you must modify the following fields in the Dialog Parameters window (see Figure 7 on page 31):

Users to grant access to

Users to grant access to must be specified when you create the system tables and when you install components. When you create the system tables it should contain all group IDs that should have access to Tivoli Decision Support for z/OS. To grant access to all users, specify PUBLIC.

When you install components, Users to grant access to should contain the group IDs that should have access to the component.

Installing Tivoli Decision Support for z/OS

SQL ID to use (in QMF)

If QMF is used with Tivoli Decision Support for z/OS in your installation, the SQL ID to use in QMF must be specified by each user. It should be one of the groups the user is connected to or the user's own user ID.

6. If you use different RACF group IDs, you can make your RACF group IDs the default for all Tivoli Decision Support for z/OS users. Edit the Tivoli Decision Support for z/OS initialization exec DRLFPROF, described in “Step 4: Preparing the dialog and updating the dialog profile” on page 27. Variables `def_syspref`, `def_othbpfx`, `def_iduser1`, and `def_idsqluser` may need to be changed, depending on the changes you made to the IDs.

Security without secondary authorization IDs

About this task

If you are not using secondary authorization IDs in DB2, the installation process is slightly different. See “Installation steps when secondary user IDs are not used” for more information.

If you are not using secondary authorization IDs in DB2, all privileges must be granted to individual users:

Procedure

1. Grant authority to the administrators:
 - a. Create all tables and views with the administrator user ID as prefix. That is, replace DRLSYS and DRL with a user ID. Only one administrator is possible.
 - b. Grant SYSADM authority to all administrators.
2. Give authority to the users in one of two ways. This is done in step 5 (see “Step 5: Setting personal dialog parameters” on page 29 for more information).
 - Specify a list of up to 8 user IDs in the field **Users to grant access to** in the **Dialog Parameters** window (Figure 7 on page 31).
 - Specify PUBLIC in the field **Users to grant access to**. This gives all users access to Tivoli Decision Support for z/OS data. This is easier to maintain than a list of user IDs.

For both cases, each user must specify his own user ID in the **SQL ID to use (in QMF)** field in the **Dialog Parameters** window, if QMF is used with Tivoli Decision Support for z/OS in your installation.

You must specify user IDs in the field **Users to grant access to** before you create the system tables. It is also used when you install components.

Installation steps when secondary user IDs are not used

Follow this example if you have several administrators. In the example, we assume that there are three administrators:

- ADMIN1 is the user who creates system tables.
- ADMIN2 and ADMIN3 are the other administrators.

When performing the installation, note these items:

- “**Step 3: Initializing the DB2 database**” on page 21
Change DRL and DRLSYS in the DRLJDBIN job to ADMIN1, ADMIN2, and ADMIN3.
- “**Step 4: Preparing the dialog and updating the dialog profile**” on page 27

No changes.

- **“Step 5: Setting personal dialog parameters” on page 29**

Use ADMIN1 as prefix for system tables, ADMIN2 and ADMIN3 as prefix for other tables. For **Users to grant access to**, specify ADMIN1, ADMIN2, ADMIN3, and all user IDs for the end users. For **SQL ID to use (in QMF)**, specify ADMIN1 (if QMF is used with Tivoli Decision Support for z/OS in your installation).

- **“Step 6: Setting up QMF” on page 32**

No changes.

- **“Step 7: Creating system tables” on page 33**

The system tables should be created with the prefix ADMIN1. Otherwise, there are no changes compared with the information in this step.

- **“Step 8: Customizing JCL” on page 35**

No changes.

- **“Step 9: Testing the installation of the Tivoli Decision Support for z/OS base” on page 36 and “Step 12: Installing components” on page 40**

If one of the secondary administrators, for example ADMIN2, wants to install the Sample component or any other component, that administrator has to change the dialog parameters before the installation to use these settings:

Prefix for system tables

ADMIN1

Prefix for other tables

ADMIN2

SQL ID to use (in QMF)

ADMIN2

When the component is installed by ADMIN2, the installed DB2 objects are created with the prefix ADMIN2.

All DB2 objects can be read by all administrators, but an object can be created only with the current administrator's primary user ID.

To make your changes the default for all Tivoli Decision Support for z/OS users, you must change the initialization exec DRLFPROF as described in “Step 4: Preparing the dialog and updating the dialog profile” on page 27.

Step 3: Initializing the DB2 database

About this task

You must use Tivoli Decision Support for z/OS to perform several DB2-related installation tasks, which are described below.

Note: Tivoli Decision Support for z/OS is an update/insert intensive DB2 application. This means that during a collect, Tivoli Decision Support for z/OS adds and updates many rows in the DB2 tables. Normal DB2 processing logs these changes. Your DB2 administrator should verify that the capacity of the DB2 logs is sufficient to cope with the increase in logging activity.

If your operational DB2 system is constrained, you might consider implementing another (analytical) DB2 system for the Tivoli Decision Support for z/OS environment.

Initializing DB2 database when installing Tivoli Decision Support for z/OS for the first time

About this task

Follow the instructions below to run the DRLJDBIN job:

Procedure

1. Copy member DRLJDBIN in the DRL182.SDRLCNTL library to the &HLQ.LOCAL.CNTL library.

You might also need to copy and customize one of the following samples: DRLJDCV0, DRLJDCVA, or DRLJDCVB. Refer to the instructions in the comments in DRLJDBIN job for more information about using these samples.

2. Modify the job card statement to run your job.
3. Customize the job for your site.

Follow the instructions in the job prolog to customize it for your site.

Note:

- a. A person with DB2 SYSADM authority (or someone with the authority to create plans, storage groups, and databases, and who has access to the DB2 catalog) must submit the job.
 - b. Do not delete steps from DRLJDBIN. Even if you have DBADM authorization, you must grant DRL and DRLSYS authority for the Tivoli Decision Support for z/OS database.
4. Submit the job to:
 - Bind the DB2 plan used by Tivoli Decision Support for z/OS.
The plan does not give privileges (it contains only dynamic SQL statements) thereby making it safe to grant access to all users (PUBLIC).
If you change the name of the plan from the default (DRLPLAN) then you must update the def_db2plan variable in DRLFPROF to specify the new plan name. You also need to modify any sample jobs that execute DRLPLC, DRL1PRE or DRLPLOGM to specify the PLAN parameter with the new plan name. Changing the plan name allows you to run versions of the TDS environment with incompatible DBRMs in the same DB2 subsystem.
 - Create the DB2 storage group and database used by Tivoli Decision Support for z/OS.
 - Grant DB2 DBADM authority as database administrators of DRLDB to DRL and DRLSYS.
 - Create views on the DB2 catalog for Tivoli Decision Support for z/OS dialog functions for users who do not have access to the DB2 catalog.

DRLJDBIN job

```
//DRLJDBIN JOB (ACCT#),'DATABASE INIT'  
//*****  
//*                                                                 *  
//* LICENSED MATERIALS - PROPERTY OF IBM                          *  
//*                                                                 *  
//* 5698-B06 Copyright IBM Corporation 1992, 2015                 *  
//* SEE COPYRIGHT INSTRUCTIONS.                                    *  
//*                                                                 *  
//*****  
//*                                                                 *  
//* NAME: DRLJDBIN                                                 *  
//*                                                                 *  
//* STATUS: Tivoli Decision support for zOS 1.8.2                 *  
//*
```

Installing Tivoli Decision Support for z/OS

```
/** *
/** FUNCTION: *
/** 1. BIND THE TDSz DB2 PLAN. *
/** 2. CREATE STORAGE GROUP AND DATABASE FOR *
/** Tivoli Decision Support for zOS 1.8.2 *
/** 3. CREATE VIEWS ON THE DB2 CATALOG. *
/** *
/** NOTES: *
/** BEFORE YOU SUBMIT THE JOB, DO THE FOLLOWING: *
/** *
/** 1. CHECK THAT THE DB2 AND TDSz DATA SET *
/** NAMES ARE CORRECT. SEARCH FOR db2loadlibrary AND *
/** DRLvrn TO FIND THE DATASET NAMES. *
/** *
/** 2. IF THE DB2 SUBSYSTEM NAME IS NOT DSN, CHANGE *
/** DSN SYSTEM(DSN) TO DSN SYSTEM(SUBSYSTEM-NAME). *
/** *
/** 3. SPECIFY A SUITABLE AUTHORIZATION ID FOR THE OWNER *
/** OPTION ON THE BIND PACKAGE AND BIND PLAN COMMANDS, *
/** OR REMOVE THIS OPTION TO USE THE AUTHORIZATION ID *
/** OF THE BINDER. *
/** *
/** 4. IF YOU WANT TO USE A PACKAGE NAME OTHER THAN DRLPLAN, *
/** CHANGE DRLPLAN IN THE BIND PACKAGE COMMAND AND IN *
/** THE PKLIST OPTION OF THE BIND PLAN COMMAND. *
/** *
/** 5. IF YOU WANT TO USE A PLAN NAME OTHER THAN DRLPLAN, *
/** CHANGE DRLPLAN IN THE BIND PLAN COMMAND AND THE *
/** GRANT EXECUTE STATEMENT TO REFER TO THE CHOSEN NAME. *
/** IF YOU CHANGE THE PLAN NAME YOU SHOULD CHANGE THE *
/** def_db2plan SPECIFICATION IN DRLFPROF, AND MODIFY *
/** THE PLAN= PARAMETER WHEREVER IT OCCURS IN SAMPLE *
/** BATCH JOBS YOU CUSTOMIZE TO USE THE NEW PLAN NAME. *
/** *
/** 6. IF YOU ARE NOT USING DB2 10.1, CHANGE DSNTIAA1 TO *
/** THE NAME OF THE CORRESPONDING PLAN FOR YOUR RELEASE. *
/** *
/** 7. IN THE CREATE STOGROUP STATEMENT, SUPPLY NAMES FOR *
/** THE VOLUME(S) AND CATALOG TO USE. *
/** IF YOU ALREADY HAVE A STORAGE GROUP DEFINED, *
/** REMOVE THE CREATE STOGROUP STATEMENT AND CHANGE *
/** THE CREATE DATABASE STATEMENT TO USE THIS STORAGE *
/** GROUP. *
/** *
/** 8. IF YOU ARE USING A DATABASE NAME THAT IS DIFFERENT *
/** FROM THE DEFAULT (DRLDB), CHANGE ALL OCCURENCES OF *
/** DRLDB TO THE NEW NAME. USE THE COMMAND: *
/** CHANGE DRLDB DATABASE-NAME WORD ALL *
/** *
/** 9. IF YOU WANT TO USE A DEFAULT BUFFER POOL FOR THE *
/** TABLE SPACES CREATED WITHIN THE DATABASE DIFFERENT *
/** FROM BPO, CHANGE THE BUFFERPOOL PARAMETER IN THE *
/** CREATE DATABASE STATEMENT AS DESIRED. *
/** *
/** 10. IF YOU WANT TO USE A DEFAULT BUFFER POOL FOR THE *
/** INDEXES CREATED WITHIN THE DATABASE, YOU CAN SPECIFY *
/** IT WITH THE ADDITIONAL PARAMETER OF THE CREATE *
/** DATABASE STATEMENT. *
/** THE PARAMETER IS INDEXBP. *
/** *
/** 11. IF YOU ARE USING A TABLE PREFIX THAT IS DIFFERENT *
/** FROM THE DEFAULT (DRL), CHANGE ALL OCCURENCES OF *
/** THE WORD DRL TO THE NEW NAME. USE THE COMMAND: *
/** CHANGE DRL TABLE-PREFIX WORD ALL *
/** *
/** 12. IF YOU ARE USING A SYSTEM TABLE PREFIX THAT IS *
/** DIFFERENT FROM THE DEFAULT (DRLSYS), CHANGE ALL *
```

Installing Tivoli Decision Support for z/OS

```
/** OCCURENCES OF DRLSYS TO THE NEW NAME. *
/** USE THE COMMAND: *
/** CHANGE DRLSYS SYSTEM-TABLE-PREFIX WORD ALL *
/** *
/** 13. IF YOU ARE USING A USER GROUP THAT IS *
/** DIFFERENT FROM THE DEFAULT (DRLUSER), CHANGE ALL *
/** OCCURENCES OF DRLUSER TO THE NEW NAME. *
/** USE THE COMMAND: *
/** CHANGE DRLUSER USER_GROUP WORD ALL *
/** *
/** 14. COPY AND CUSTOMIZE THE REQUIRED DRLJDCVx MEMBER *
/** USING STEPS 6, 9, 10 AND 11 ABOVE. *
/** IF NO CUSTOMIZATION IS REQUIRED THEN THE *
/** DRLJDCVx MEMBER DOES NOT NEED TO BE COPIED. *
/** FOR DB2 VERSION 10 USE DRLJDCVA. *
/** FOR DB2 VERSION 11 USE DRLJDCVB. *
/** UNCOMMENT THE DD STATEMENT REFERRING TO THE *
/** REQUIRED MEMBER AND MODIFY IT IF NECESSARY *
/** TO REFER TO THE COPIED AND CUSTOMIZED VERSION. *
/** *
/** CHANGE ACTIVITY: *
/** 01 2001-08-24 BB - PQ49985 : *
/** ADDED NOTE (POINT 11) IN ORDER TO *
/** NOTIFY THAT, FOR MIGRATIONS, *
/** EXISTING OBJECTS NEED BE DROPPED *
/** BEFORE CREATE *
/** 02 2002-05-27 BB - PQ61494 : *
/** CHANGED NOTE (POINT 11) IN ORDER TO *
/** NOTIFY THAT, FOR MIGRATIONS, *
/** EXISTING OBJECTS NEED BE UNLOADED *
/** BEFORE DROP; ALSO, AFTER CREATE, *
/** THEY NEED BE RELOADED. *
/** 03 2003-03-07 RV - SPECIFY THAT THE COMMENTS ADDED BY *
/** APARS PQ49985 AND PQ61494 REFER TO *
/** MIGRATION TO A NEW RELEASE OF DB2 *
/** PTR537 *
/** *
/** CHANGE ACTIVITY: *
/** CHANGE FLAG TYPE DATE DESCRIPTION *
/** -----*
/** $D0=DCR066, TDS180,01/06/07,ADL(SM): Update TDS Version and *
/** DB2 dataset names. *
/** $00=PK54663,TDS180,15/10/07,ADL(MG): Add CCSID EBCIDC to *
/** CREATE DATABASE and Encoding*
/** EBCDIC to BIND PLAN *
/** $D1=DCR116, TDS181,15/05/09,ADL(RC): Update TDS Version *
/** $01=PM07437,TDS181,05/02/10,ADL(SY): Add comments for *
/** using a different plan name *
/** $02=PI26707,TDS181,06/10/14,ADL(SY): *
/** Make views compatible with DB2 V10 and V11. *
/** $D2=RW110662,TDS182,21/01/15,ADL(AA): TDS 182, DB2 V10.V11 *
/** $D3=RW127254,TDS182,02/03/15,ADL(SY): Bind to package *
/** *
/** *****
/** DBINIT EXEC PGM=IKJEFT01
/** STEPLIB DD DISP=SHR,DSN=db2loadlibrary
/** DBRMLIB DD DISP=SHR,DSN=DRLvrn.SDRLCNTL(DRLPSQLX)
/** SYSPRINT DD SYSOUT=*
/** SYSTSPRT DD SYSOUT=*
/** SYSTSIN DD *
DSN SYSTEM(DSN)
BIND PACKAGE(DRLPLAN) OWNER(authid) MEMBER(DRLPSQLX) -
ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC)
BIND PLAN(DRLPLAN) OWNER(authid) PKLIST(*.DRLPLAN.*) -
ACTION(REPLACE) RETAIN

RUN PROGRAM(DSNTIAD) PLAN(DSNTIAA1) -
```



```

        LIB('DSNA10.RUNLIB.LOAD')
    END
//SYSIN      DD *

GRANT EXECUTE ON PLAN DRLPLAN TO PUBLIC;

CREATE STOGROUP DRLSG
    VOLUMES (VOLUME1, VOLUME2, ...)
    VCAT CATALOG-NAME;

GRANT USE OF STOGROUP DRLSG TO DRL, DRLSYS WITH GRANT OPTION;

CREATE DATABASE DRLDB
    BUFFERPOOL BP0
    CCSID EBCDIC
    STOGROUP DRLSG;

GRANT DBADM ON DATABASE DRLDB TO DRL, DRLSYS WITH GRANT OPTION;

CREATE VIEW DRLSYS.DRLTABLES AS
    SELECT * FROM SYSIBM.SYSTABLES
        WHERE CREATOR IN ('DRL','DRLSYS');
GRANT SELECT ON DRLSYS.DRLTABLES TO DRL, DRLUSER;

CREATE VIEW DRLSYS.DRLCOLUMNS AS
    SELECT * FROM SYSIBM.SYSCOLUMNS
        WHERE TBCreator IN ('DRL','DRLSYS');
GRANT SELECT ON DRLSYS.DRLCOLUMNS TO DRL, DRLUSER;

CREATE VIEW DRLSYS.DRLTABAUTH AS
    SELECT * FROM SYSIBM.SYSTABAUTH
        WHERE TCreator IN ('DRL','DRLSYS');
GRANT SELECT ON DRLSYS.DRLTABAUTH TO DRL;

CREATE VIEW DRLSYS.DRLINDEXPART AS
    SELECT * FROM SYSIBM.SYSINDEXPART
        WHERE IXCreator IN ('DRL','DRLSYS');
GRANT SELECT ON DRLSYS.DRLINDEXPART TO DRL;

CREATE VIEW DRLSYS.DRLKEYS AS
    SELECT * FROM SYSIBM.SYSKEYS
        WHERE IXCreator IN ('DRL','DRLSYS');
GRANT SELECT ON DRLSYS.DRLKEYS TO DRL;

/**
/** Customize and use DRLJDCVA for DB2 V10 *only*
/**
/**      DD DISP=SHR,DSN=DRLvrm.SDRLCNTL(DRLJDCVA)
/**
/** Customize and use DRLJDCVB for DB2 V11 *only*
/**
/**      DD DISP=SHR,DSN=DRLvrm.SDRLCNTL(DRLJDCVB)
/**

```

Customizing the CICS Partitioning feature

About this task

If you are going to use the CICS Partitioning feature, run the DRLJDBIP job. DRLJDBIP creates additional storage groups that are used in the partitioned table spaces of the CICS Partitioning feature.

To run DRLJDBIP:

Installing Tivoli Decision Support for z/OS

Procedure

1. Copy member DRLJDBIP in the DRL182.SDRLCNTL library to the &HLQ.LOCAL.CNTL library.
2. Modify the job card statement to run your job.
3. Customize the job for your site. Follow the instructions in the job prolog.
4. Submit the job.

Note: A person with DB2 SYSADM authority (or someone who has access to the DB2 catalog) must submit the job.

DRLJDBIP job

DRLJDBIP job (member of DRL182.SDRLCNTL)

```
//DRLJDBIP JOB (ACCT#),'SG FOR PARTITION'                                00010000
//*****                                                                    00020000
//*                                                                           *      00030000
//* LICENSED MATERIALS - PROPERTY OF IBM                                 *      00040000
//*                                                                           *      00050000
//* 5698-B06 Copyright IBM Corporation 1992, 2015                       *      00060000
//* SEE COPYRIGHT INSTRUCTIONS.                                         *      00070000
//*                                                                           *      00080000
//*****                                                                    00090000
//*                                                                           *      00100000
//* NAME: DRLJDBIP                                                       *      00110000
//*                                                                           *      00120000
//* STATUS: Tivoli Decision Support for zOS 1.8.2                       *      00130000
//*                                                                           *      00140000
//* FUNCTION:                                                             *      00150000
//*   CREATE STORAGE GROUPS FOR PARTITIONING                            *      00220800
//*                                                                           *      00221000
//* NOTES:                                                                 *      00221100
//*   YOU NEED DB2 SYSADM AUTHORITY TO SUCCESSFULLY EXECUTE           *      00221200
//*   THIS JOB. BEFORE YOU SUBMIT THE JOB, DO THE FOLLOWING:          *      00221300
//*                                                                           *      00221500
//*   1. CHECK THAT THE DB2 AND TDSz DATA SET                          *      00221600
//*       NAMES ARE CORRECT. SEARCH FOR db2loadlibraru AND              *      00221700
//*       DRLvrn TO FIND THE DATA SET NAMES.                           *      00221800
//*                                                                           *      00221900
//*   2. IF THE DB2 SUBSYSTEM NAME IS NOT DSN, CHANGE                  *      00222000
//*       SYSTEM=DSN TO SYSTEM=SUBSYSTEM-NAME IN THE SYSTSIN          *      00222100
//*       STEP.                                                         *      00222200
//*                                                                           *      00222700
//*   3. IF YOU ARE USING A SYSTEM TABLE PREFIX THAT IS              *      00222800
//*       DIFFERENT FROM THE DEFAULT (DRLSYS), CHANGE DRLSYS           *      00222900
//*       IN THE GRANT STATEMENT TO THE NEW NAME.                       *      00223000
//*                                                                           *      00223100
//*   4. IF YOU ARE USING A TABLE PREFIX THAT IS DIFFERENT           *      00223200
//*       FROM THE DEFAULT (DRL), CHANGE DRL IN THE GRANT               *      00223300
//*       STATEMENT TO THE NEW NAME.                                     *      00223400
//*                                                                           *      00223500
//*   5. IF YOU ARE USING STORAGE GROUP DIFFERENT FROM THE            *      00223600
//*       DEFAULT, CHANGE STOGR1, STOGR2, STOGR3, STOGR4,              *      00223700
//*       IN THE CREATE AND GRANT STATEMENTS. IF YOU NEED              *      00223800
//*       MORE/LESS STORAGE GROUPS MODIFY THE NUMBER (ADD/              *      00223900
//*       DELETE) OF CREATE AND GRANT STATEMENTS.                       *      00224000
//*                                                                           *      00224100
//*   6. SPECIFY THE VOLUMES YOU ARE USING FOR THE STORAGE             *      00224200
//*       GROUPS (VOLSTOGRxx).                                          *      00224300
//*                                                                           *      00224400
//*   7. SPECIFY THE VCAT PARAMETER IN THE CREATE STATEMENT            *      00224500
//*       (CATALOG-NAME)                                                *      00224600
//*                                                                           *      00224700
//*   8. IF YOU ARE NOT USING DB2 10.1, CHANGE DSNTIAA1 TO             *      00224800
//*       THE NAME OF THE CORRESPONDING PLAN FOR YOUR RELEASE.         *      00224900
```

Installing Tivoli Decision Support for z/OS

```
//* * 00225000
//* CHANGE ACTIVITY: * 00419000
//* CHANGE FLAG TYPE DATE DESCRIPTION * 00420000
//* -----* 00430000
//* $D0=DCR066, TDS180,01/06/07,ADL(SM): Update TDS Version and * 00440000
//* DB2 dataset names. * 00450000
//* $D1=DCR116, TDS181,15/05/09,ADL(RC): Update TDS Version * 00460000
//* $D2=RW110662 TDS182,21/01/15,ADL(AA): Update DB2 version * 00470000
//* $D3=RW127659 TDS182,11/03/15,ADL(AA): Clean up comments * 00470100
//* * 00470200
/****** 00470300
/* 00470400
//DBSTRG EXEC PGM=IKJEFT01 00470500
//STEPLIB DD DISP=SHR,DSN=db2loadlibrary 00470600
//DBRMLIB DD DISP=SHR,DSN=DRLvrm.SDRLCNTL(DRLPSQLX) 00470700
//SYSPRINT DD SYSOUT=* 00470800
//SYSTSPRT DD SYSOUT=* 00470900
//SYSTSIN DD * 00471000
DSN SYSTEM(DSN) 00471100
RUN PROGRAM(DSNTIAD) PLAN(DSNTIAA1) - 00472000
LIB('DSNA10.RUNLIB.LOAD') 00473000
END 00474000
//SYSIN DD * 00700000
CREATE STOGROUP STOGR1 00750000
VOLUMES (VOLSTOGR11, VOLSTOGR12, ...) 00760000
VCAT CATALOG-NAME; 00770000
00780000
GRANT USE OF STOGROUP STOGR1 TO DRL, DRLSYS WITH GRANT OPTION; 00790000
00790100
CREATE STOGROUP STOGR2 00790200
VOLUMES (VOLSTOGR21, VOLSTOGR22, ...) 00790300
VCAT CATALOG-NAME; 00790400
00790500
GRANT USE OF STOGROUP STOGR2 TO DRL, DRLSYS WITH GRANT OPTION; 00790600
00790700
CREATE STOGROUP STOGR3 00790800
VOLUMES (VOLSTPGR31, VOLSTOGR32, ...) 00790900
VCAT CATALOG-NAME; 00791000
00792000
GRANT USE OF STOGROUP STOGR3 TO DRL, DRLSYS WITH GRANT OPTION; 00793000
00793100
CREATE STOGROUP STOGR4 00793200
VOLUMES (VOLSTOGR41, VOLSTOGR42, ...) 00793300
VCAT CATALOG-NAME; 00793400
00793500
GRANT USE OF STOGROUP STOGR4 TO DRL, DRLSYS WITH GRANT OPTION; 00793600
00793800
/* 00863600
```

Step 4: Preparing the dialog and updating the dialog profile

About this task

The load library and the exec library must be allocated at the startup of your TSO logon procedure. Tivoli Decision Support for z/OS dynamically allocates other libraries and data sets as it starts, and allocates others as certain functions are performed. This step describes how to set up procedures for startup and for allocating the libraries and data sets that Tivoli Decision Support for z/OS needs.

Ensure that the load library, exec library, DB2 load library, QMF load library (optional), GDDM libraries, and load libraries, are accessible to your TSO session

Installing Tivoli Decision Support for z/OS

Procedure

1. Make the load library (DRL182.SDRLOAD), DB2 load library, QMF load library, and the GDDM load library accessible by performing one of these tasks:

- a. Allocate the SDRLOAD library, DB2 load library (SDSNLOAD), QMF load library (SDSQLOAD), and the GDDM load library (SADMMOD) to STEPLIB in the generic logon procedure

```
//STEPLIB DD DISP=SHR,DSN=DRL182.SDRLOAD
//          DD DISP=SHR,DSN=QMF710.SDSQLOAD
//          DD DISP=SHR,DSN=GDDM.SADMMOD
//          DD DISP=SHR,DSN=DSN710.SDSNLOAD
```

- b. Add SDRLOAD, SDSQLOAD, SADMMOD, and SDSNLOAD to the link list.
- c. Copy SDRLOAD, SDSQLOAD, SADMMOD, and SDSNLOAD members to a library already in the link list. Make sure that the DB2 modules DSNALI, DSNHLI2, and DSNTIAR are linked in 31-bit addressing mode

2. Make the local exec library and the Tivoli Decision Support for z/OS exec library (DRL182.SDRLEXEC) accessible by performing one of these tasks:

- a. Allocate the libraries to SYSPROC in the logon procedure. For example:

```
//SYSPROC DD DISP=SHR,DSN=&HLQ.LOCAL.EXEC
//          DD DISP=SHR,DSN=DRL182.SDRLEXEC
```

- b. Allocate the libraries to SYSEXEC in the logon procedure. For example:

```
//SYSEXEC DD DISP=SHR,DSN=&HLQ.LOCAL.EXEC
//          DD DISP=SHR,DSN=DRL182.SDRLEXEC
```

- c. Use the ATLIB function to allocate the libraries.

If Tivoli Decision Support for z/OS is invoked by using the ATLIB function on the application level, make sure that only the Tivoli Decision Support for z/OS exec library is included. Allocate other exec libraries to user level by using the **ATLIB ACT USER(EXEC)** command.

3. Make the ADMPC data set accessible by allocating it in the logon procedure. For example:

```
//ADMPC DD DISP=SHR,DSN=GDDM.SADMPCF
```

Tivoli Decision Support for z/OS dynamically allocates other libraries and data sets, such as the GDDM symbols data set GDDM.SADMSYM, when a user starts a dialog. "Allocation overview" on page 69 describes the libraries that Tivoli Decision Support for z/OS allocates and when it allocates them.

4. If you have used any values other than default values for DRLJDBIN or for Tivoli Decision Support for z/OS data set names, you must modify the *userid.DRLFPROF* file (allocated copying the DRLFPROF member of DRL182.SDRLCNTL).

DRLEINI1 sets dialog defaults for all users. Tivoli Decision Support for z/OS stores defaults for each user in member DRLPROF in the library allocated to the ISPPROF ddname, which is usually *tsoprefix.ISPF.PROFILE*. Edit DRLFPROF to include default values so users do not need to change dialog parameter fields.

5. Allocate a sequential data set with name user.DRLFPROF, LRECL=80 BLKSIZE=32720 RECFM=FB and copy the DRLFPROF member of the SDRLCNTL library.
6. Locate and change any variable values that you have changed during installation.

Note:

- Change values for data set names that identify DB2 and, optionally, QMF and GDDM libraries.
- If you do not use QMF with Tivoli Decision Support for z/OS, change the value for `qmfuse` to NO.
- If you do not use GDDM with Tivoli Decision Support for z/OS, change the value for `gddmuse` to NO. (If QMF is used, GDDM must be used.)

“Modifying the DRLFPROF dataset” on page 55 shows the DRLFPROF file containing the parameters to be modified.

“Overview of the Dialog Parameters window” on page 56 shows the administration dialog window and the default initialization values that DRLFPROF sets.

“Dialog parameters - variables and fields” on page 58 describes parameters and shows the interrelationship of DRLEINI1 and the Dialog Parameters.

7. You can add Tivoli Decision Support for z/OS to an ISPF menu by using this ISPF statement:

```
CMD(%DRLEINIT) [DEBUG] [RESET] [DBRES] [REPORTS | R] [ADMINISTRATION | A]
```

To access a dialog from the command line of an ISPF window, any authorized user can issue the command **TSO %DRLEINIT** from the command line of an ISPF window.

The optional `DEBUG` parameter sets on a REXX trace for the initialization execs. This helps you solve problems with data set and library allocation.

The optional `RESET` parameter sets the ISPF profile variables to their default value. It has the same effect as deleting the DRLPROF member from the local (ISPPROF) profile library.

The optional `REPORTS` parameter takes you directly to the reporting dialog. You can abbreviate this to R.

The optional `ADMINISTRATION` parameter takes you directly to the administration dialog. You can abbreviate this to A.

Step 5: Setting personal dialog parameters

About this task

If you have edited the dialog parameters profile, file DRLFPROF from the DRL182.SDRLCNTL library, and copied it into the sequential data set `userid.DRLFPROF` in “Step 4: Preparing the dialog and updating the dialog profile” on page 27 to match your installation values, you do not need to follow the instructions in this step to change the parameters unless you want to use the reporting dialog in administrator mode

Authorized administrators can use the reporting dialog in administrator mode to view or modify all reports. Otherwise, a reporting dialog user uses the dialog in end-user mode, the default. In this mode, a user can view only public and privately-owned reports. In end-user mode, a user can modify only reports he or she created.

Tivoli Decision Support for z/OS stores parameters for each user in member DRLPROF in the library allocated to the ISPPROF ddname, which is usually `tsoprefix.ISPF.PROFILE`

This topic describes the procedure for the Tivoli Decision Support for z/OS dialogs if you *did not* edit the DRLFPROF file. Perform this step if necessary.

Installing Tivoli Decision Support for z/OS

To set dialog parameters:

Procedure

1. From the command line of an ISPF/PDF window, do one of the following:
 - a. Type `TSO %DRLEINIT` to display the **Tivoli Decision Support for z/OS Primary Menu** (Figure 5).
 - OR -
 - b. Type `TSO %DRLEINIT Administration` to display the **Administration** window (Figure 3 on page 10).

Note: Reporting dialog users can access the **Dialog Parameters** window from the **Options** pull-down of the **Primary Menu** or the **Reports** window.

```
Options Help
-----
                    Tivoli Decision Support for z/OS Primary Menu

Select one of the following. Then press Enter.

      2  1. Reports
         2. Administration

Command ==> _____
F1=Help   F2=Split   F3=Exit   F9=Swap   F10=Actions F12=Cancel
```

Figure 5. Tivoli Decision Support for z/OS Primary Menu

2. If you start from the **Primary Menu**, type `2Administration`, and press **Enter** to display the **Administration** window (see Figure 3 on page 10)
3. From the **Administration** window, select 1, System, to display the System window (Figure 6 on page 31)

Note: If your installation does not use QMF, Import QMF initialization query is not selectable.

4. From the **System** window, select 1, **Dialog parameters**.

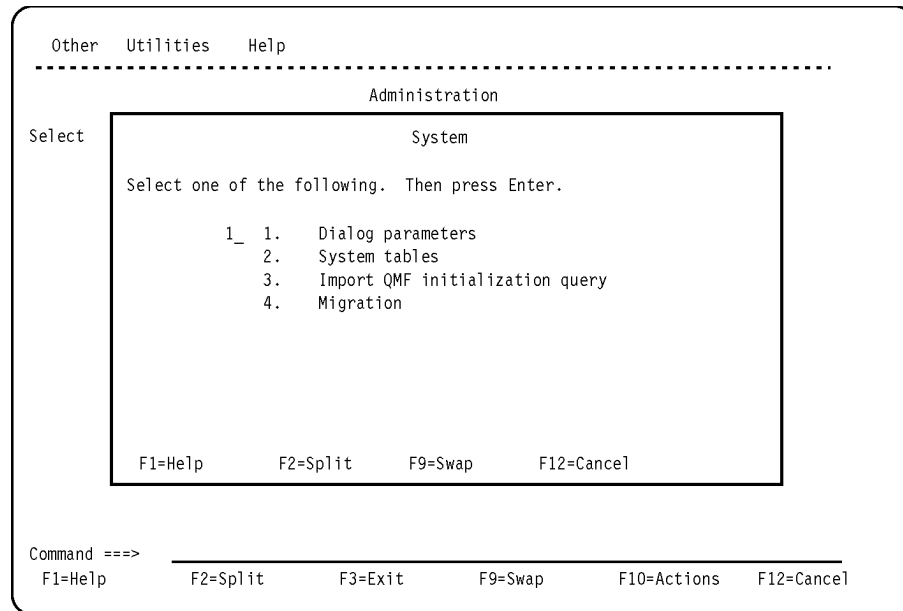


Figure 6. System window - Option 1

Note: If your installation does not use QMF with Tivoli Decision Support for z/OS, the contents of this window is slightly different from what you see here. Both versions of the Dialog Parameters window are shown in “Overview of the Dialog Parameters window” on page 56.

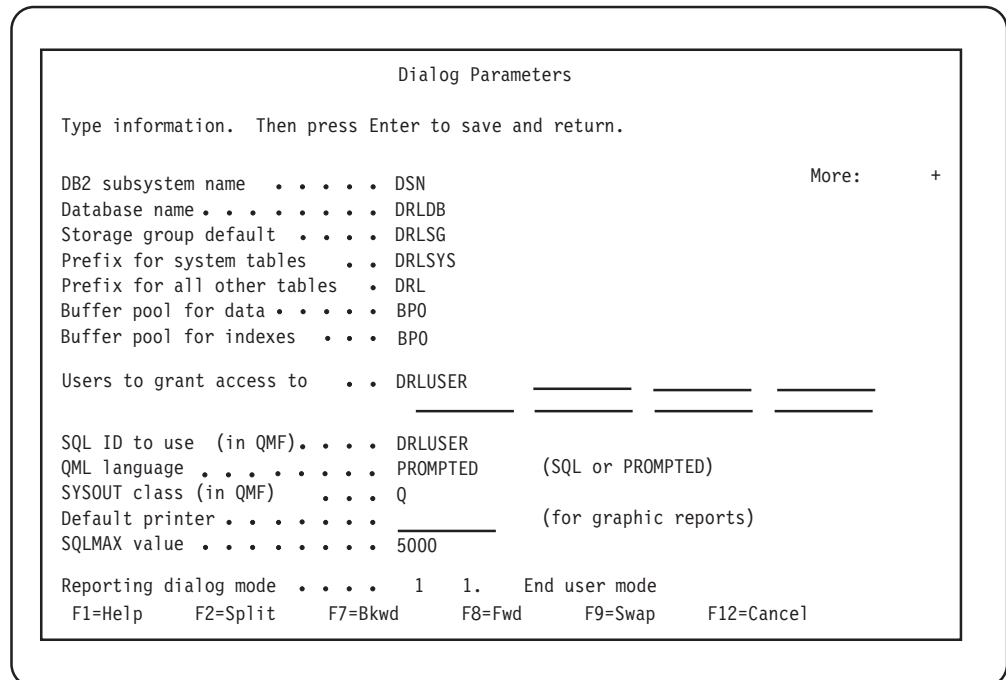


Figure 7. Dialog Parameters window

Note: When you see this indicator:

More: +

Installing Tivoli Decision Support for z/OS

in the upper-right corner of a Tivoli Decision Support for z/OS window, press **F8** to scroll down. If the indicator shows a minus sign (-), press **F7** to scroll up. For more information about using Tivoli Decision Support for z/OS dialog windows, refer to the description in the *Guide to Reporting*.

You must scroll through the window to display all its fields. “Overview of the Dialog Parameters window” on page 56 shows the entire **Dialog Parameters** window, both the version shown if QMF is used with Tivoli Decision Support for z/OS and the version shown if QMF is not used with it. “Dialog parameters - variables and fields” on page 58 has a description of the fields in the window.

5. Make modifications and press **Enter**

Changes for administration dialog users and for end users are the same. You must identify the correct names of any data sets (including prefixes and suffixes) that you changed from default values during installation.

Tivoli Decision Support for z/OS saves the changes and returns to the **System** window. Although some changes become effective immediately, all changes become effective in your next session when Tivoli Decision Support for z/OS can allocate any new data sets you may have selected.

Step 6: Setting up QMF

About this task

Note: Tivoli Decision Support for z/OS can use QMF, for example, to display and work with reports. If your installation does not use QMF, the information in this topic does not apply, and option 3, **Import QMF** initialization query, is not selectable in the **System** window.

When Tivoli Decision Support for z/OS starts QMF, it runs a query, (DRLQINIT), to set the current SQL ID (by default, DRLUSER) that gives users required authority in QMF and lets them access objects in the QMF lists.

To import the QMF query from member DRLQINIT (in the DRL182.SDRLDEFS library) and save it in QMF as DRLSYS.DRLQINIT, from the **System** window (Figure 8 on page 33), select 3, **Import QMF initialization query**, and press **Enter**.

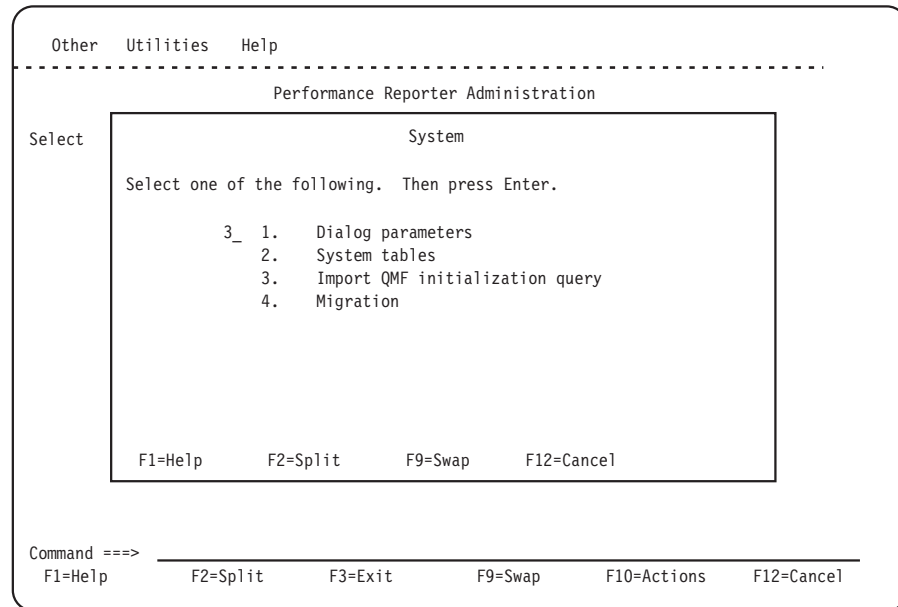


Figure 8. System window - Option 3

Tivoli Decision Support for z/OS imports the query into QMF and then returns to the **System** window.

Step 7: Creating system tables

About this task

Before you can use all dialog functions, you must create the DB2 tables. These DB2 tables are used by Tivoli Decision Support for z/OS to store its definitions and are known as *system tables*.

To create system tables follow these steps:

Procedure

1. From the **System** window, select **2, System tables**.
The **System Tables** window is displayed. (Figure 9 on page 34).

Installing Tivoli Decision Support for z/OS

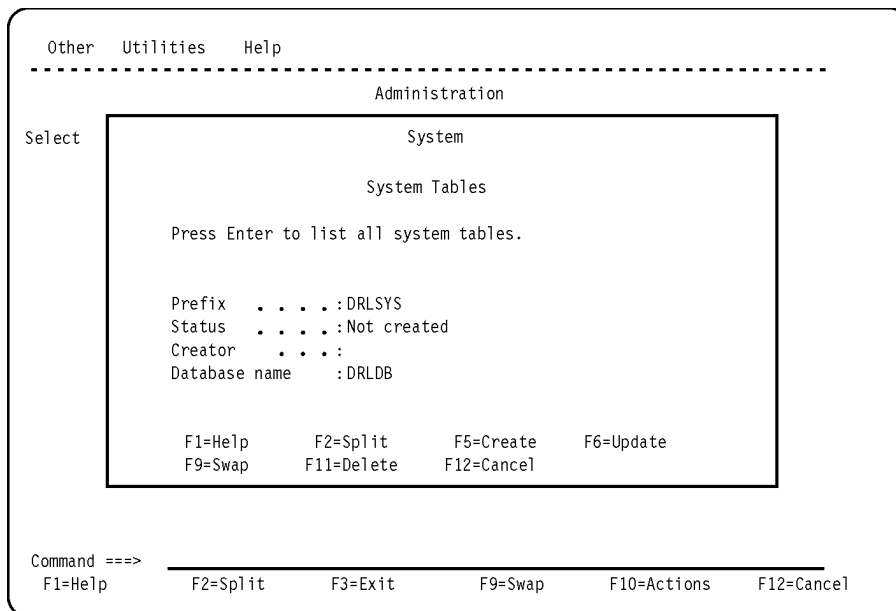


Figure 9. System Tables (not created) window

2. Press **F5 (Create)**. Tivoli Decision Support for z/OS creates system tables and fills in information about feature components by searching DRL182.SDRLDEFS to see which features you have installed with SMP/E.

Tivoli Decision Support for z/OS displays messages in a browse window, if a problem has occurred. In this case, look for errors at the beginning of the listing. Resolve any errors such as this:

```
DSNT408I  SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN
        UNAVAILABLE RESOURCE. REASON 00D70025, TYPE OF RESOURCE 00000220
        AND RESOURCE NAME DB2A.DSNDBC.DRLDB.A.I0001.A001
```

For information about specific DB2 messages, refer to the *Messages and Problem Determination*, SH19-6902. System messages should be error free, with a DB2 return code of zero. After creating the system tables, Tivoli Decision Support for z/OS returns to the System Tables window where you must press F12 to return to the System window.

During the process of creating system tables, these administrative reports are also created:

- PRA001 - INDEXSPACE cross-reference. For more information, see “PRA001 - Indexspace cross-reference” on page 281.
- PRA002 - ACTUAL TABLESPACE allocation. For more information, see “PRA002 - Actual tablespace allocation” on page 282.
- PRA003 - TABLE PURGE condition. For more information, see “PRA003 - Table purge condition” on page 284.
- PRA004 - LIST COLUMNS for a requested table with comments. For more information, see “PRA004 - List columns for a requested table with comments” on page 285
- PRA005 - LIST ALL TABLES with comments. For more information, see “PRA005 - List all tables with comments” on page 286
- PRA006 - LIST USER MODIFIED objects. For more information, see “PRA006 - List User Modified Objects” on page 287

Creating and updating system tables with a batch job

About this task

You can also create, update, and delete Tivoli Decision Support for z/OS system tables by running TSO/ISPF in batch mode. Sample job DRLJCSTB shows an example of how to submit a request to program DRLEAPST to create system tables. You can update or delete system tables by passing a different request to DRLEAPST, as described in the comments in DRLJCSTB.

The TSO/ISPF batch job step must include:

- DRLFPROF DD referring to your DRLFPROF data set
- ISPPROF DD referring to a PDS with RECFM=F and LRECL=80. If you have made changes to the Tivoli Decision Support for z/OS dialog parameters and have not also made those changes in your DRLFPROF data set, then the ISPPROF DD should refer to your ISPF profile data set and you should not specify the RESET parameter to DRLEINIT.
- ISPPLIB, ISPMLIB, ISPSLIB, and ISPTLIB DDs referring to your Tivoli Decision Support for z/OS and ISPF panel, message, skeleton, and table data sets.
- ISPLOG DD referring to a data set with RECFM=VA and LRECL=125.
- SYSTSIN DD referring to instream data, or a data set, containing a command to invoke DRLEINIT, for example:

```
IPSTART CMD(%DRLEINIT RESET)
```

- DRLBIN (batch input) DD referring to instream data or a data set containing a command to invoke DRLEAPST with a request to perform the required function, for example:

```
DRLEAPST CREATE
```

DRLEAPST is the only program that can be invoked in this way.

Step 8: Customizing JCL

About this task

The DRL182.SDRLCNTL library contains several batch jobs that you can copy to &HLQ.LOCAL.CNTL and customize. Customization includes inserting correct data set names and the correct DB2 subsystem ID. These jobs, described in Chapter 6, “Setting up operating routines,” on page 85, are:

DRLJBATR

A sample job for printing and saving all (or a selected subset) of the batch reports. See “Using job DRLJBATR to run reports in batch” on page 113 for more information.

DRLJCOLL and DRLJCOxx

A sample job for collecting log data. See “Collecting log data” on page 85 for more information.

DRLJCOPY

A sample job for backing up a Tivoli Decision Support for z/OS tablespace with the DB2 COPY utility. See “Backing up the Tivoli Decision Support for z/OS database” on page 105 for more information.

DRLJDICT

A sample job for partitioning the CICS_DICTIONARY table, if the CICS

Installing Tivoli Decision Support for z/OS

Partitioning feature is going to be used. See the CICS Partitioning feature chapter in *CICS Performance Feature Guide and Reference* for more information.

DRLJEXCE

A sample job for producing Tivoli Information Management for z/OS problem records. See “Administering problem records” on page 121 for more information.

DRLJEXCP

A sample job for partitioning the EXCEPTION_T table, if the CICS Partitioning feature is going to be used. See the CICS Partitioning feature chapter in *CICS Performance Feature Guide and Reference* for more information.

DRLJPURG

A sample job for purging data from the database. See “Purge utility” on page 104 for more information.

DRLJREOR

A sample job for reorganizing the Tivoli Decision Support for z/OS database with the DB2 REORG utility. See “Purge utility” on page 104 for more information.

DRLJRUNS

A sample job for updating statistics on Tivoli Decision Support for z/OS tablespaces with the DB2 RUNSTATS utility. See “Monitoring the size of the Tivoli Decision Support for z/OS database” on page 108 for more information.

DRLJTBSR

A sample job for producing a detailed report about the space required for all, or a subset of, a selected component’s tables. See “Understanding table spaces” on page 96 for more information.

If you already have jobs for maintaining DB2, for example, COPY, REORG or RUNSTATS, you can continue to use them for this purpose, instead of using the Tivoli Decision Support for z/OS jobs.

Step 9: Testing the installation of the Tivoli Decision Support for z/OS base

About this task

Before you install Tivoli Decision Support for z/OS feature components, ensure that the installation has been successful:

Procedure

1. Install the Sample component using the information in “Installing a component” on page 126. Although editing lookup tables is a usual part of online component installation, you need not edit the sample lookup table to successfully complete this test. For a description of what is provided with the sample component, see Chapter 13, “Sample components,” on page 253
2. After you install the Sample component, select **3, Logs**, from the **Administration** window and press **Enter**.
The Logs window is displayed (Figure 10 on page 37).

```

Log Utilities View Other Help
-----
                                Logs                                ROW 1 TO 1 OF 1

Select a log. Then press Enter to display record definitions.

/ Logs                Description
/ SAMPLE              Sample log definition
***** BOTTOM OF DATA *****

Command ==>
F1=Help   F2=Split   F3=Exit   F5=Log def  F6=Datasets  F7=Bkwd
F8=Fwd    F9=Swap    F10=Actions  F11=Collect  F12=Cancel
    
```

Figure 10. Logs window

- From the **Logs** window, select the **SAMPLE** log and press **F11**.
The Collect window is displayed.

```

Log Utilities View Other Help
+-----+
                                Collect
+-----+

Type information. Then press Enter to edit the collect JCL.

Data set  DRLxxx.SDRLEDFS(DRLSAMPL)                (reqd)
Volume . . _____ (If not cataloged)
Unit . . . _____ (Required for batch if Volume defined)

Reprocess . . . . . 2  1. Yes
                                     2. No
Commit after . . . . . 1  1. Buffer full
                                     2. End of file
                                     3. Specify number of records

Number of records . . _____
Buffer size . . . . . 10
Extention . . . . . 2  1. K
                                     2. M

Condition . . . . . _____ >
F1=Help   F2=Split   F4=Online  F5=Include  F6=Exclude
F9=Swap   F10=Show fld F11=Save def F12=Cancel

+-----+
Command ==>
F1=Help   F2=Split   F3=Exit   F5=Log def  F6=Datasets  F7=Bkwd
F8=Fwd    F9=Swap    F10=Actions  F11=Collect  F12=Cancel
+-----+
    
```

Figure 11. Sample log statistics output

- Type **DRLxxx.SDRLEDFS(DRLSAMPL)** in the **Data set** field and press **F4**.
The online collect is started. When it finishes, it displays statistics about the data it collected.
- Press **F3** to return to the **Logs** window after you finish looking at the messages.
- Press **F3** to return to the **Administration** window
- From the **Administration** window, select **5, Reports**, and press **Enter**.

Installing Tivoli Decision Support for z/OS

The **Reporting Dialog Defaults** window is displayed. (Refer to *Guide to Reporting* for more information.)

- Press **Enter** to display the Reports window.

```
Report Batch Group Search Options Other Help
                                Reports                Row 1 to 9 of 9

Select a report. Then press Enter to display.

Group . . . . . : All reports

/ Report                                     ID
ACTUAL TABLESPACE SPACE allocation         PRA002
INDEXSPACE cross-reference                 PRA001
List all tables with comments              PRA005
List columns for a requested table with comments PRA004
List User Modified Objects                 PRA006
/ Sample Report 1                           SAMPLE01
Sample Report 2                             SAMPLE02
Sample Report 3                             SAMPLE03
TABLE PURGE Condition                       PRA003
***** Bottom of data *****

Command ==>
F1=Help    F2=Split  F3=Exit   F4=Groups  F5=Search  F6=Listsrch
F7=Bkwd    F8=Fwd    F9=Swap   F10=Actions F11>Showtype F12=Cancel
```

Figure 12. Reports window

- From the **Reports** window, select **Sample Report 1**. Type a character other than a question mark in the selection field and press **Enter**.
The **Data Selection** window is displayed.

```
                                Data Selection          ROW 1 TO 1 OF 1

Type values. Then press Enter to generate the report.

Report ID . . . . : Sample Report 1

Variable          Value                                     Oper  Req
SYSTEM_ID         _____ > + = No
***** BOTTOM OF DATA *****

Command ==> _____
F1=Help    F2=Split  F4=Prompt  F5=Table   F6=Chart   F7=Bkwd
F8=Fwd     F9=Swap   F10=Showfld F11=Hdrval F12=Cancel
```

Figure 13. Data Selection window

- Press **Enter** to generate the report.
The query associated with the report is run and the report is displayed through GDDM/ICU³. (Figure 107 on page 256 shows the report.)

3. If your installation does not have GDDM, the report is displayed in tabular format.

11. When you finish viewing the report, press **F9** to exit from GDDM/ICU, and press **F3** (Exit) to return to the **Reports** window.
12. From the **Reports** window, press **F3** to return to the **Administration** window.

Step 10: Reviewing DB2 parameters

About this task

Before you install components, you can review DB2 table and index space parameters such as:

- Buffer pool.
- Compression.
- Erase on deletion.
- Free space.
- Lock size.
- Number of partitions, for a partitioned space.
- Number of subpages, for an index space.
- Primary and secondary space.
- Segment size.
- Type of space.
- VSAM data set password.

Please give careful consideration to these parameters, as they can affect the performance of your system.

Note: Before you assign a buffer pool to a component index or table space, activate the buffer pool and add the USE privilege to the privilege set for the buffer pool.

To change parameters:

Procedure

1. From the **Administration** window, select **2, Components**, and press **Enter**.
2. Select a component.
3. Select the **Space** pull-down.
4. Select **Tablespaces**, to change table space definitions, or select **Indexes**, to change index definitions.

What to do next

If you are unsure about the meaning of a field, press **F1** to get help. For more information, refer to the CREATE INDEX and CREATE TABLESPACE command descriptions in the *DB2 Universal Database for OS/390 and z/OS: SQL Reference*

Tivoli Decision Support for z/OS saves the changed definitions in your local definitions library. When you save a changed definition, it tells you where it is saving it, and prompts you for a confirmation before overwriting a member with the same name.

Step 11: Determining partitioning mode and keys

About this task

Some component definitions use the GENERATE statement to create the tables, spaces, partitioning, and indexes. The table space, partitioning, and index attributes can easily be changed by updating the appropriate profile in the GENERATE_PROFILES and GENERATE_KEYS system tables.

Procedure

1. Consult the guide for the component you are installing. Many will have a job which must be run to set up Store groups, partition ranges or keys. Follow the instructions for that component before proceeding. If the component does not support Generated Table spaces and Indexes, you may skip this step.
2. When using GENERATE TABLESPACE the type of table space created is determined by the TABLESPACE_TYPE field in the GENERATE_PROFILES system table.
3. If you decide to use Range Partitioned table spaces TABLESPACE_TYPE=RANGE, you will need to adjust the range values in the GENERATE_KEYS system table.

What to do next

The supplied values for these tables are in the member DRLTKEYS in the SDRLEDFS data set, and the tables are created and loaded during the creation of the Tivoli Decision Support for z/OS System Tables. These values may be reviewed prior to creating the Tivoli Decision Support for z/OS system tables. If changes are required, you may make a copy in your userid.LOCAL.DEFS data set and make the required changes prior to System Table creation.

Alternatively, once loaded into the System Tables these values may be changed by various methods:

- Using the Tivoli Decision Support for z/OS table edit facility.
- Using SQL UPDATE statements. For example, to change the TABLESPACE_TYPE from the supplied value of RANGE to GROWTH for IMS the statement would look like the following example:

```
SQL UPDATE <sysprefix>.DELPART_PROFILES
        SET TABLESPACE_TYPE='GROWTH'
        WHERE PROFILE='IMS';
```

Step 12: Installing components

About this task

In previous installation steps, you have:

- Installed all Tivoli Decision Support for z/OS data sets.
- Set up access to Tivoli Decision Support for z/OS data.
- Initialized the Tivoli Decision Support for z/OS database.
- Allocated the required data sets of related products.
- Initialized Tivoli Decision Support for z/OS dialog parameters.
- Created Tivoli Decision Support for z/OS system tables.
- Initialized QMF for Tivoli Decision Support for z/OS (if applicable).
- Customized sample Tivoli Decision Support for z/OS JCL.

- Tested Tivoli Decision Support for z/OS (with the Sample component).
- Determined partitioning mode and keys.
- Reviewed DB2 parameters.

You are now ready to install Tivoli Decision Support for z/OS features. To install components, use the information in “Installing a component” on page 126, and in these books:

Feature name	Book name
AS/400 Performance	<i>AS/400 System Performance Feature Guide and Reference</i>
CICS Performance	<i>CICS Performance Feature Guide and Reference</i>
Distributed Systems Performance	<i>Distributed Systems Performance Feature Guide and Reference, SH19-4018</i>
IMS Performance	<i>IMS Performance Feature Guide and Reference</i>
Network Performance	<i>Network Performance Feature Installation and Administration</i>
System Performance	<i>System Performance Feature Reference Volume I and II</i>

To install Resource Accounting for z/OS (part of the base function), see the *Resource Accounting for z/OS, SH19-4495* book.

Installing the Usage and Accounting Collector

About this task

The CIMS Lab Mainframe collector is incorporated into Tivoli Decision Support and called the Usage and Accounting Collector.

For a description of the Usage and Accounting Collector, see “System Overview” in the *Usage and Accounting Collector User Guide*.

To install the Usage and Accounting Collector, follow these steps:

Procedure

1. “Customizing the Usage and Accounting Collector” on page 42
2. “Allocating and initializing Usage and Accounting files” on page 44

Results

To verify your installation, follow these steps:

1. “Processing SMF data using DRLNJOB2 (DRLCDATA and DRLCACCT)” on page 45.
2. “Running DRLNJOB3 (DRLCMONY) to create invoices and reports” on page 47.
3. “Processing Usage and Accounting Collector Subsystems” on page 49.

To support programs such as CICS, DB2, IDMS, IMS, VM/CMS, VSE, DASD Space Chargeback, and Tape Storage Accounting, edit and run the appropriate jobs. Examples of member names are DRLNCICS, DRLNDB2, DRLNDISK.

Installing Tivoli Decision Support for z/OS

To check your SMP/E installation, see “Step 1: Reviewing the results of the SMP/E installation” on page 17.

Customizing the Usage and Accounting Collector

About this task

Installation job DRLNINIT invokes the REXX program DRLCINIT. This program is a utility that customizes Usage and Accounting Collector jobs to your specifications. DRLCINIT inserts job cards, adds high level nodes to all Usage and Accounting Collector data sets, changes VOLSER numbers, and specifies DSCB model names.

Run job DRLNINIT and follow these steps:

Procedure

1. Replace sample job card with user job card.
2. Insert or replace data set name high-level qualifiers.
3. Insert serial numbers on the VOLUME parameter.
4. Insert DSCB model names.

Note: If you do not run DRLCINIT, you must change each job member manually as you use it.

DRLNINIT

About this task

To execute job DRLNINIT, follow these instructions:

Procedure

1. DRL.SDRLCNTL (DRLMFLST) contains the list of Usage and Accounting Collector jobs that are used in this utility.
2. The SMP/E process allocates &HLQ.LOCAL.CNTL. This DSN stores the customized jobs. The Usage and Accounting Collector JCL is copied to this library and changes are made in this library. The first step in DRLNINIT performs the copy. This makes it possible to execute DRLNINIT repeatedly until the desired result is achieved.

Replace the two occurrences of &HLQ.LOCAL.CNTL in DRLNINIT with the filename that was allocated during the SMP/E install.

3. Job card replacement.

A standard job card can be inserted with a unique jobname. The following parameters in STEP020 control the job card replacement:

JCDSN=

Specifies the file containing the standard job card.

For example: JCDSN=DRL.SDRLCNTL(JBCARD) The contents in member JBCARD is used as the job card.

JCLINES=

The number of lines to use from JCDSN.

For example: JCLINES=2 The first two lines in the JCDSN member are used as a job card.

JCMASK=

A unique job name can be generated for the execution jobs. The

JCMASK is used to specify the common part of the jobname and the position of a sequential number. After the first character, you must enter a sequence of '*' (asterisk) characters to indicate where to insert the job sequence number. The sequence mask is from 2 to 6 characters in length:

Examples:

JCMASK

Jobnames generated

DRL****

DRL0001, DRL0002, DRL0003...

P***Q**

P000001Q, P000002Q, P000003Q...

DRLDRL**

DRL01DRL, DRL02DRL, DRL03DRL...

JCSKIP=

Specify any non-blank character and the Job card replacement process will be skipped.

For example: JCSKIP=Y No job card customization of the Usage and Accounting Collector execution jobs is done.

4. Insert or replace data set name high level qualifiers. The default filenames used for the Usage and Accounting Collector files start with the high-level qualifier of 'DRL'. The HLQ process in the DRLCINIT utility allows this default to be replaced or an additional high-level qualifier to be inserted. The following parameters in STEP020 control the HLQ processing:

HLQACT=

Specifies the action to perform: R=Replace, I=Insert.

For example: HLQACT=R Every occurrence of a filename with the high-level qualifier of 'DRL.', will be replaced with the value in HLQDSN.

HLQDSN=

The new value to use for the high-level qualifier.

For example: HLQDSN=DRL.TDSZUAC The default filenames are changed to start with 'DRL.TDSZUAC'.

HLQSKIP=

Specify any non-blank character and the HLQ processing is skipped.

For example: HLQSKIP=Y

No customization of the Usage and Accounting Collector data set names is done.

5. Insert VOLSER numbers. At various places within the Usage and Accounting Collector jobs, volume serial numbers are needed. The DRLCINIT job allows you to replace them all globally. The default volume serial numbers are "?????" throughout the JCL. The default volume serial appears in IDCAMS processing as VOL(?????) and VOL=SER=????? and is used for VSAM file allocation. The JCL also uses VOL=SER=????? for temporary space allocations. The following parameters in STEP020 control the VOLSER processing:

VOL= The replacement volume serial to use instead of "?????"

Installing Tivoli Decision Support for z/OS

VSSKIP=

Specify any non-blank character and the VOLSER processing is skipped.

For example: VSSKIP=Y

No customization of the Usage and Accounting Collector VOL or VOL=SER parameters is done.

6. Insert DSCB model names.

A model DSCB parameter is used for the proper functioning of Generation Data Groups (GDGs). The Usage and Accounting Collector JCL is distributed with all model DSCB preferences set to 'MODELDCB'. If your installation does not require the use of this parameter, you can delete it manually from the JCL. The DSCB processing can be used to change the default to a value used at your installation. The following parameters in STEP020 control the DSCB processing:

MDDSCB=

The replacement model DSCB to use instead of MODELDCB.

MDSKIP=

Specify any non-blank character and the model DSCB processing will be skipped.

For example: MDSKIP=Y

No customization of the Usage and Accounting Collector model DSCB will be done.

The DRLCINIT utility produces statistics for the execution. If any exceptions are noted, they can be found listed in the DRLMXCEP member of &HLQ.LOCAL.CNTL. These exceptions might or might not be severe enough to cause a JCL error; check DRLMXCEP if exceptions are reported.

Statistic report DDNAME SYSTSPRT

Processing.....

Completed SYSTSIN

69 Files

0 Exceptions

JobCard : 68 Replacements

HLQ : 1389 Replacements

Volume : 30 Replacements

ModelDSCB: 207 Replacements

Normal completion

Allocating and initializing Usage and Accounting files

About this task

DRLNJOB1 is a member in DRL182.SDRLCNTL. This job creates four permanent files and four Generation Data Groups (GDGs). The permanent files are:

Usage and Accounting Collector client

Member DRLMCLNT contains sample client records. For information about client records, see "Client Identification and Budget Reporting – DRLCCLNT and DRLCBDGT" in the *Usage and Accounting Collector User Guide*.

Rate Members DRLMRATE, DRLMRT01, and DRLMRT02 contain sample Rate

records. For information about rate records, see “Computer Center Chargeback Program – DRLCMONY” in the *Usage and Accounting Collector User Guide*.

Dictionary

Members DRLKxxxx contain the default record definitions for the Usage and Accounting Collector Dictionary. For more information about the Usage and Accounting Collector Dictionary, see “Dictionary – CIMSDTVS” in the *Usage and Accounting Collector User Guide*.

Status and Statistics VSAM

The Status and Statistics file is a VSAM file that should be allocated so that checkpoint and statistical information can be recorded for program DRLCEXTR. Use the default values to create the VSAM files.

Note: You do not need to set rates or identify clients at this time.

For the JCL, see member DRLNJOB1 in DRL182.SDRLCNTL.

Processing SMF data using DRLNJOB2 (DRLCDATA and DRLCACCT)

About this task

This job, which is divided into two steps, runs programs DRLCDATA and DRLCACCT. These programs interface with the z/OS-SMF data set and create the DRL.DRLCACCT.DAILY batch chargeback file.

DRLNJOB2 job is the basis for daily processing and is the only job required on a daily basis for batch chargeback. Logically, it is run immediately after the SMF data set is unloaded to disk or tape. After DRLNJOB2 processing is finished, data set DRL.DRLCACCT.DAILY contains z/OS batch and TSO accounting records, and data set DRL.SMF.HISTORY contains reformatted SMF records.

Note: It is recommended that you read “SMF Interface Program – DRLCDATA” and “Accounting File Creation Program – DRLCACCT” in the *Usage and Accounting Collector User Guide* before you start changing the default control statements.

Procedure

1. OB STEP DRLC2A

This executes program DRLCDATA. For more information, see “SMF Interface Program – DRLCDATA” in the *Usage and Accounting Collector User Guide*.

Table 2. Explanation of Program DRLCDATA

Input/output	DDNAME	Description
INPUT	SMFIN	This is the SMF DUMP data set.
INPUT	CIMSCNTL	Data set DRL182.SDRLCNTL (DATAINPT) Contains input control statements. For more information, see the Control Statement Table in Chapter 2. “SMF Interface Program – DRLCDATA” in the <i>Usage and Accounting Collector User Guide</i> .

Installing Tivoli Decision Support for z/OS

Table 2. Explanation of Program DRLCDATA (continued)

Input/output	DDNAME	Description
OUTPUT	CIMSSMF	Usage and Accounting Collector reformatted SMF data set. Contains each SMF record from the input data set unless limited by a records statement. This data set is designed as a backup data set of reformatted SMF Records. Depending on installation requirements, you might choose to DD DUMMY this data set, or to COMMENT the statement.
OUTPUT	CIMSACCT	This data set contains selected SMF chargeback records (6, 30, 101, 110). This data set is used as input in step DRLC2B.
OUTPUT	CIMSCICS	This data set contains CICS records (SMF Type 110). This record is used by the Usage and Accounting Collector CICS interface programs.
OUTPUT	CIMSDB2	This data set contains DB2 records (SMF Type 101). This record is used by the Usage and Accounting Collector DB2 interface programs.

2. SMF Merge

It is recommended that you insert a merge between steps DRLC2A and DRLC2B to create a history of data set DRL.SMF.HISTORY (see member DRLNSMFM in DRL182.SDRLCNTL). The merge field is 7 for one character. Use a cartridge tape and block the output data set to 32K (BLKSIZE = 32760).

The Usage and Accounting Collector Merge is a sample SORT/MERGE set of JCL that creates a sorted history data set of Usage and Accounting Collector accounting records can be found in data set DRL182.SDRLCNTL member DRLNMERG. This job should be run daily after the batch and online Usage and Accounting Collector jobs have been executed.

If DRLNMERG is done on a daily basis, at the end of the month, the Usage and Accounting Collector master file is in account code sort sequence. You should maintain the history data sets on tape. Leave the daily files on disk for daily reports and set up generation data sets to tape for the history file.

3. JOB STEP DRLC2B

This executes program DRLCACCT, which processes the data set created by program DRLCDATA (DDNAME CIMSACCT) and generates the Usage and Accounting Collector batch chargeback data set. For details, see "Accounting File Creation Program – DRLCACCT" in the *Usage and Accounting Collector User Guide*.

Table 3. Explanation of Program DRLCACCT

Input/output	DDNAME	Description
INPUT	CIMSDATA	Reformatted SMF records. These records are created by DDNAME CIMSACCT in program DRLCDATA. The Usage and Accounting Collector Suspense file for unmatched job step and print records is appended to DDNAME CIMSDATA.
INPUT	CIMSCNTL	Control statements.

Table 3. Explanation of Program DRLCACCT (continued)

Input/output	DDNAME	Description
INPUT	CIMSTABL	Optional user-supplied table to convert job names and/or job card account codes to a new format. For more information, see Chapter 3. "Accounting File Creation Program – DRLCACCT" in the <i>Usage and Accounting Collector User Guide</i> .
INPUT	CIMSDTVS	Usage and Accounting Collector Dictionary VSAM file.
INPUT	CIMSPDS	Control statements. This data set is used by DRLCACCT when PROCESS CIMS SERVER RESOURCE RECORDS control statement is specified. A member, DRLMALSA, in this data set contains the control members for the different records.
OUTPUT	IMSACT2	Usage and Accounting Collector batch chargeback file containing the 79x accounting records. This data set is used by DRLCEXTR and DRLCMONY.
OUTPUT	CIMSUSPN	Suspense file. This data set contains Step and Print records that have not been matched with a Job Start or Job Stop record.
OUTPUT	CIMSEXCP	This data set contains records that have not been matched with entries in the CIMSTABL data set.
OUTPUT	CIMSPRNT	This data set contains the runtime parameters and the results of the run.
OUTPUT	CIMSMMSG	This data set contains informational messages.
OUTPUT	CIMSEL	Usage and Accounting Collector accounting records. This data set contains the records that failed date selection when the PROCESS CIMS MAINTENANCE and NON-SELECTED FILE PROCESSING ON control statements are specified.
OUTPUT	CIMSUNSP	Unsupported CSR records. This data set contains all CSR records that did not have a definition within CIMSDTVS.

Note: For JCL information, see member DRLNJOB2 in DRL182.SDRLCNTL.

Running DRLNJOB3 (DRLCMONY) to create invoices and reports

About this task

DRLNJOB3 contains the JCL to run program DRLCMONY, which creates invoices and zero-cost invoices (rate determination).

Installing Tivoli Decision Support for z/OS

Billing control statements are contained in member DRLMMNY. Edit these statements to customize Usage and Accounting Collector for your installation.

You can use the Usage and Accounting Collector defaults as distributed until you decide on client information, billing rates, and control information.

To run DRLNJOB3, follow these steps:

Procedure

1. Run DRLN3A.
This step converts the 79x accounting records into CSR+ records. DRLCMONY supports only CSR+ records.
2. Run DRLC3B.
This step sorts the data set created by step DRLC3A into account code, job name, and job log number sequence.
3. Run DRLC3C.
This step is for the Computer Center Billing System – DRLCMONY.

Input/output	DDNAME	Description
INPUT	CIMSACCT	Integrated chargeback data set.
INPUT	CIMSCLVS	Client records.
INPUT	CIMSCNTL	Control statements.
INPUT	CIMSRTVS	Billing rates.
INPUT	CIMSCLDR	Usage and Accounting Collector calendar file.
INPUT	CIMSNCPU	CPU normalization statements.
INPUT	CIMSSCPU	CPU job class and priority surcharge statements.
OUTPUT	SYSOUT	Messages
OUTPUT	CIMSPRNT	Processing results.
OUTPUT	CIMSINVC	Invoices.
OUTPUT	CIMSMSG	Informational messages.
OUTPUT	CIMSSUM	Summary records by account. One record per account and billable item – (Rate Code).
OUTPUT	CIMSIDNT	Identifier data that can be loaded into a Tivoli Usage and Accounting Manager database. This file is produced by DRLCMONY in Server mode.
OUTPUT	CIMSDETL	Detail data that can be loaded into a Tivoli Usage and Accounting Manager database. This file is produced by DRLCMONY in server mode.
OUTPUT	CIMSUMRY	Summary data that can be loaded into a Tivoli Usage and Accounting Manager database. This file is produced by DRLCMONY in server mode.

For record descriptions, see Appendix. “Accounting File Record Descriptions” in the *Usage and Accounting Collector User Guide*.

For JCL information, see member DRLNJOB3 in DRL182.SDRLCNTL.

Processing Usage and Accounting Collector Subsystems

About this task

Note: This step is optional.

Usage and Accounting Collector is now installed and ready to be customized for batch chargeback. After you are comfortable with the results you are receiving from the Usage and Accounting Collector z/OS batch system, you can start integrating data from the wide range of subsystems that Usage and Accounting Collector supports.

To integrate a Usage and Accounting Collector subsystem, perform the following steps:

Procedure

1. Edit the appropriate JCL member. For example, DRLNCICS.
2. Create an account code conversion table.
3. Process the job.
4. Merge the output with the input to program DRLCMONY (DRLNJOB3).
5. Run DRLNJOB3 to generate the integrated invoices.

Results

The following list provides a list of member names for some of the most commonly-used Usage and Accounting Collector subsystems.

Table 4. Usage and Accounting Collector Subsystem Member Names (Partial List)

Subsystem Member name	Description
DRLNCICS	CICS Support
DRLNDB2	DB2
DRLNMQSR	MQSeries®
DRLNDISK	DASD Space
DRLNTAPE	Tape Storage
DRLNIMS	IMS
DRLNUNIV	ROSCOE, ADABAS/SMF, IDMS/SMF, RJE, WYLBUR, Oracle, MEMO, Control-T, BETA

Installing multiple Tivoli Decision Support for z/OS systems

About this task

You can install more than one Tivoli Decision Support for z/OS system on the same DB2 subsystem. This is useful if you want to develop and test new Tivoli Decision Support for z/OS applications.

Note: You cannot use DB2 Copy to copy the objects from the first installation to the new one. If you do, QMF definitions may be lost.

To install another Tivoli Decision Support for z/OS system, repeat the installation from “Step 2: Setting up security” on page 18 to “Step 12: Installing components” on page 40 and specify different values for:

Installing Tivoli Decision Support for z/OS

- DB2 subsystem
- Database
- System table prefix
- Other tables prefix
- RACF groups (if necessary)
- Local data sets

For example, assume your user ID is BILL, and you want a private Tivoli Decision Support for z/OS system.

Dialog parameter
Value

DB2 subsystem
DB2T

Database
BILLDB

System table prefix
BILL

Other table prefix
BILL

Users to grant access to
BILL

Local data sets
BILL.DEFS....and so on

Other users cannot use this system because BILL is not a DB2 secondary authorization ID nor a RACF group ID. If you want to share this new Tivoli Decision Support for z/OS system, establish a valid RACF group ID and use the group ID as the prefix instead of BILL.

Installing Tivoli Decision Support for z/OS features separately

About this task

Use this information if you are installing Tivoli Decision Support for z/OS features separately; that is, not at the same time as you installed the Tivoli Decision Support for z/OS base and any features. To install features follow these steps.

Procedure

1. Follow the instructions in the *Tivoli Decision Support for z/OS Program Directory* to use SMP/E to install all the performance features required. If you have already installed a feature with SMP/E, you need not install it again unless you are reinstalling to correct a previous installation error.
2. Update Tivoli Decision Support for z/OS system tables with information about the features you are installing
 - a. From the Tivoli Decision Support for z/OS **Administration** window, select 1, System, to display the System window (Figure 6 on page 31)
 - b. Select 2, System Tables, to display the System Tables window (Figure 14 on page 51).

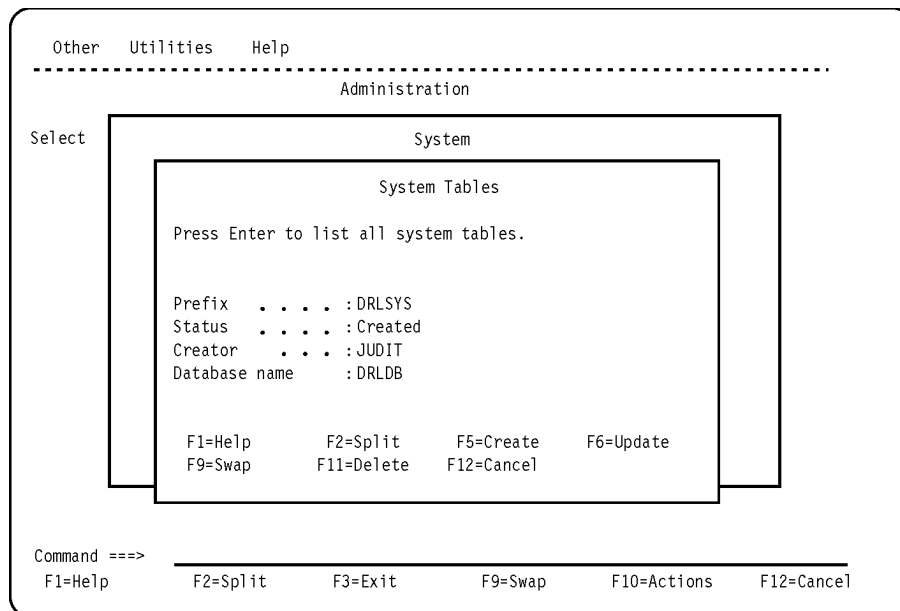


Figure 14. System Tables (created) window

- c. Press F6 (Update) to update the system tables with information about the newly installed features.
- d. Resolve any DB2 errors that appear at the top of the browse window. The successful installation of a component that is a part of the feature (described in “Installing a component” on page 126) verifies the feature's installation

Installing Tivoli Decision Support for z/OS

Part 2. Installation reference

Chapter 3. Dialog parameters

This topic describes dialog parameters that are set initially by member DRLEINI1 in the DRLxxx.SDRLEXEC library and read from the *userid*.DRLFPROF data set. Tivoli Decision Support for z/OS initializes a new user's first dialog session with parameter settings from *userid*.DRLFPROF. From that point forward, a user's dialog parameters are in personal storage in member DRLPROF in the library allocated to the ISPPROF ddname, which is usually *tsoprefix*.ISPF.PROFILE. If DRLFPROF exists, a user changes parameter values through the Dialog Parameters window. DRLEINI1 continues to set parameters that do not appear in the Dialog Parameters window. It does this when a user starts Tivoli Decision Support for z/OS.

“Step 4: Preparing the dialog and updating the dialog profile” on page 27 describes the installation step where *userid*.DRLFPROF is customized for your site. It refers to this section for descriptions of:

- “Modifying the DRLFPROF dataset”
- “Overview of the Dialog Parameters window” on page 56
- “Dialog parameters - variables and fields” on page 58
- “Allocation overview” on page 69

Modifying the DRLFPROF dataset

About this task

The DRLFPROF data set contains user modifiable parameters. A sample of the DRLFPROF data set is provided in member DRLFPROF in library SDRLCNTL. To customize DRLFPROF with your site specific values, allocate a data set with the name *userid*.DRLFPROF and copy in the sample DRLFPROF member from the SDRLCNTL library.

For a description of the fields that can be modified in the *userid*.DRLFPROF data set, see “Dialog parameters - variables and fields” on page 58.

When editing the *userid*.DRLFPROF data set, note that:

- Tivoli Decision Support for z/OS regards any characters after the /* characters as comments. This means that /* JCL comments cannot be used. A closing */ is recommended but not required.
- The format for field assignment is: field-name = value [/* comment [*/]] except as noted below. No other tokens may be present. Tokens are case insensitive.
- Each field assignment must be completed on one line. Continuation is not supported.
- Any value (even integer values) can be given as a REXX-style string, delimited by the single (') or double (") quotation marks. Escaping of delimiter characters works in the same way as a REXX string.
- If a value does not begin with a ' or " character, only the first blank-separated word present after the = character is taken.
- Though sequence numbering in DRLFPROF may not cause errors, it is not supported and should be turned off.
- For the fields DEF_JCLSTA1, DEF_JCLSTA2, DEF_JCLSTA3 and DEF_JCLSTA4, the value is taken as any characters between the = and the '/*, or end of the line

Modifying the DRLFPROF dataset

if no comment is present. Delimiting this value with double quotation marks (") is highly recommended but not required.

- If the above recommendations are adhered to, the DRLFPROF file syntax is a subset of REXX syntax and so syntax highlighting can be used for easier editing.

Overview of the Dialog Parameters window

The parameters displayed in the Dialog Parameters window depend on whether your installation uses QMF. This section shows the parameters used when QMF is used. For an overview of the parameters used when QMF is not installed on your system, refer to Figure 16 on page 58.

Dialog Parameters when QMF is used

Figure 15 on page 57 is a logical view of the Dialog Parameters window, which is available from the System window of the administration dialog and from the Other pull-down of the reporting dialog. You can change the personal settings that control their your dialog sessions. For a description of the fields in this window, see "Dialog parameters - variables and fields" on page 58.

Overview of the Dialog Parameters window

```
Dialog Parameters

Type information. Then press Enter to save and return.

More: +

DB2 subsystem name . . . . . DSN
DB2 plan name for TDS . . . . . DRLPLAN
Database name . . . . . DRLDB
Storage group default . . . . . DRLSG
Prefix for system tables . . . . . DRLSYS
Prefix for all other tables . . . . . DRL
Show TDS environment data . . . . . NO (YES or NO)

Buffer pool for data . . . . . BP0
Buffer pool for indexes . . . . . BP0

Users to grant access to . . . . . DRLUSER
_____

SQL ID to use (in QMF) . . . . . DRLUSER
QMF language . . . . . PROMPTED (SQL or PROMPTED)
SYSOUT class (in QMF) . . . . . Q
Default printer . . . . . _____ (for graphic reports)
SQLMAX value . . . . . 5000

Reporting dialog mode . . . . . 1 1. End user mode
                               2. Administrator mode

Dialog language . . . . . 1 1. English
                               3. Japanese

DB2 data sets
  Prefix . . . . . DSN810
  Suffix . . . . . _____
QMF data sets prefix . . . . . QMF810

Tivoli Decision Support for z/OS data sets prefix . . . . . DRL181
Temporary data sets prefix . . . . . (user_ID substituted)

Local definitions data set . . . . . DRL.LOCAL.DEFS
Local GDDM formats data set . . . . . DRL.LOCAL.ADMCFORM
Local messages data set . . . . . DRL.LOCAL.MESSAGES
Saved reports data set . . . . . DRL.LOCAL.REPORTS
Saved charts data set . . . . . DRL.LOCAL.CHARTS

Job statement information (required for batch jobs):
//(user_ID substituted) JOB (000000,XXXX),'USER1',MSGLEVEL=(1,1),
// NOTIFY=(user_ID substituted),MSGCLASS=Q,CLASS=E,REGION=4096K
//*

F1=Help   F2=Split   F7=Bkwd   F8=Fwd    F9=Swap   F12=Cancel
```

Figure 15. Dialog Parameters window, when QMF is used

Dialog Parameters when QMF is not used

Figure 16 on page 58 is a logical view of the Dialog Parameters window, which is available from the System window of the administration dialog and from the Other pull-down of the reporting dialog. Tivoli Decision Support for z/OS users can change personal settings that control their dialog sessions. For a description of the fields in this window, see “Dialog parameters - variables and fields” on page 58.

Dialog parameters - variables and fields

```
Dialog Parameters

Type information. Then press Enter to save and return.

More: +

DB2 subsystem name . . . . . DSN
DB2 plan name for TDS . . . . . DRLPLAN
Database name . . . . . DRLDB
Storage group default . . . . . DRLSG
Prefix for system tables . . . DRLSYS
Prefix for all other tables . DRL
Show TDS environment data . . NO (YES or NO)

Buffer pool for data . . . . . BP0
Buffer pool for indexes . . . . BP0

Users to grant access to . . DRLUSER _____
                                     _____
                                     _____
                                     _____

Batch print SYSOUT class . . A
Printer line count per page 60
SQLMAX value . . . . . 5000

Reporting dialog mode . . . . 1 1. End user mode
                               2. Administrator mode

Dialog language . . . . . 1 1. English
                               2. Japanese

DB2 data sets
  Prefix . . . . . DB2.V820
  Suffix . . . . .

TDS for zOS data
sets prefix . . . . . TDS182
Temporary data sets prefix (user_ID substituted)

Local defs data set . . . . . DRL.LOCAL.DEFS
Local User defs data set . . DRL.LOCAL.USER.DEFS
Local GDDM formats data set DRL.LOCAL.ADMCFORM
Local messages data set . . . DRL.LOCAL.MESSAGES
Saved reports data set . . . DRL.LOCAL.REPORTS
Saved charts data set . . . . DRL.LOCAL.CHARTS

Job statement information (required for batch jobs):

//(user_ID substituted) JOB (000000,XXXX),'USER1',MSGLEVEL(1,1),
//  NOTIFY=&SYSUID,MSGCLASS=Q,CLASS=E,REGION=4096K
//*
//*
F1=Help    F2=Split    F7=Bkwd    F8=Fwd     F9=Swap    F12=Cancel
```

Figure 16. Dialog Parameters window, when QMF is not used

Dialog parameters - variables and fields

Most variable names in *userid.DRLFPROF* and field names in the Dialog Parameters window are directly related. The following table describes the relationship between the variables and fields and describes how Tivoli Decision Support for z/OS uses the values to allocate libraries or control other dialog functions. It also describes variables and fields that do not have exact equivalents.

“Modifying the DRLFPROF dataset” on page 55 shows the user-modifiable area of the file that is processed at the product startup. The “Overview of the Dialog Parameters window” on page 56 shows the Dialog Parameters window. “Allocation overview” on page 69 describes the data sets allocated by Tivoli Decision Support for z/OS.

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
modtenu	N/A	None	
The fully qualified name of the user tables' library, if any. The maximum supported value is 99999999.			
db2plib2	N/A	SDSNPFPP	
The DB2 panel library, which, depending on the value of db2def, is either a fully qualified name or a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.			
db2plibe	N/A	SDSNPFPE	
The English DB2 panel library, which, depending on the value of db2def, is either a fully qualified name or a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.			
db2plibk	N/A	SDSNPFPK	
The Japanese DB2 panel library, which, depending on the value of db2def, is either a fully qualified name or a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.			
qmfprint	N/A	YES	
Specifies whether the QMF output is saved in the DSQPRINT data set (YES) or in the SYSOUT class (NO).			
def_db2subs	DB2 subsystem name	DSN	
The DB2 subsystem where Tivoli Decision Support for z/OS resides. This required field can be 4 alphanumeric characters. The first character must be alphabetic. The default value is DSN. If the value in this field is something other than DSN, it was changed during installation to name the correct DB2 subsystem. Do not change the value to name another DB2 subsystem to which you might have access. Tivoli Decision Support for z/OS must use the DB2 subsystem that contains its system, control, and data tables.			
def_db2plan	DB2 plan name for TDS	DRLPLAN	
The DB2 plan name to which the distributed Tivoli Decision Support for z/OS for z/OS DBRM has been bound. This required field can be 8 alphanumeric characters. The first character must be alphabetic. The default value for this field is DRLPLAN. If the value in this field is something other than DRLPLAN, it may have been changed during installation to refer to a customized plan name for Tivoli Decision Support for z/OS . Only change the plan name shown here if instructed to do so by your Tivoli Decision Support for zOS system administrator.			
def_dbname	Database name	DRLDB	
The DB2 database that contains all Tivoli Decision Support for z/OS system, control, and data tables. The value of this field is set during installation. This required field can be up to 8 alphanumeric characters. The first character must be alphabetic. The value of this field depends on the naming conventions at your site. The default database is DRLDB. If this value is something other than DRLDB, it is likely the default value for your site. Do not change this name to identify another DB2 database to which you have access. You must use the DB2 database that contains Tivoli Decision Support for z/OS.			
def_storgrp	Storage group default	DRLSG	
The storage group that Tivoli Decision Support for z/OS uses for the DB2 database identified in the Database name field. This required field can be 8 alphanumeric characters. The first character must be alphabetic. The default is DRLSG. If the value of the field is something other than DRLSG, it was changed during installation. Do not change the value of this field to another storage group to which you might have access; Tivoli Decision Support for z/OS uses the value of this field to create new tables.			
def_syspref	Prefix for system tables	DRLSYS	

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
<p>The prefix of all Tivoli Decision Support for z/OS system and control DB2 tables. The value of this field depends upon your naming conventions and is determined during installation.</p> <p>This required field can be 8 alphanumeric characters. The first character must be alphabetic.</p> <p>The default is DRLSYS. If the value is something other than DRLSYS, it was changed during installation.</p> <p>Do not change the value; Tivoli Decision Support for z/OS uses this value to access its system tables.</p>			
def_othtbpfx	Prefix for all other tables	DRL	
<p>The prefix of Tivoli Decision Support for z/OS data tables in the DB2 database.</p> <p>Valid values are determined at installation.</p> <p>This required field can be 8 alphanumeric characters. The first character must be alphabetic.</p> <p>The default is DRL. If the value is something other than DRL, it was changed during installation.</p>			
def_drlshwid	Show TDS environment data	NO	
<p>Specifies whether or not to display the Tivoli Decision Support for z/OS environment data in the main panels.</p> <p>This required field can have a value of YES or NO.</p> <p>The default value for this field is NO.</p>			
def_tsbpool	Buffer pool for data	BP0	
<p>The default buffer pool for Tivoli Decision Support for z/OS tablespaces. This field can have values from BP0 to BP49, from BP8K0 to BP8K9, from BP16K0 to BP16K9, from BP32K to BP32K9. The buffer pool implicitly determines the page size. The buffer pools BP0, BP1, ..., BP49 hold 4-KB pages. The buffer pools BP8K0, BP8K1, ..., BP8K9 hold 8-KB pages. The buffer pools BP16K0, BP16K1, ..., BP16K9 hold 16-KB pages. The buffer pools BP32K, BP32K1, ..., BP32K9 hold 32-KB pages.</p>			
def_ixbpool	Buffer pool for indexes	BP0	
<p>The default buffer pool for Tivoli Decision Support for z/OS indexes. This field can have values from BP0 to BP49 (The buffer pool for indexes must identify a 4-KB buffer pool).</p>			
def_iduser1	Users to grant access to	DRLUSER	
<p>The user IDs or group IDs of users who are granted DB2 access to the next component you install. Users or user groups with DB2 access to a component have access to the tables and views of the component. You can specify up to 8 users or group IDs in these fields.</p> <p>You must specify a value for at least one of the fields.</p> <p>Each user ID or group ID can be 8 alphanumeric characters. The first character must not be numeric.</p> <p>The default is DRLUSER, as shipped by IBM. You can use any user group ID that is valid for your DB2 system. You should use one such group ID to define a list of core Tivoli Decision Support for z/OS users (who might include yourself). It is a good idea to leave such a core group as the value in one of the fields, regardless of whether you control user access to various components by adding other group IDs.</p> <p>You can grant users access to the tables and views of a component by listing them here before you install the component.</p> <p>Consider using RACF group IDs or DB2 secondary authorization IDs and specifying them in these fields before installing a component. It is easier to connect individual user IDs to an authorized group than it is to grant each individual access to each table or view that they need.</p>			
def_idsqlusr	SQL ID to use (in QMF)	DRLUSER	
<p>This field is used only if your installation uses QMF.</p> <p>The DB2 primary or secondary authorization ID to which you are connected. Tivoli Decision Support for z/OS uses the value of this field in the SET CURRENT SQLID as it starts QMF. The ID must have DB2 authorization to Tivoli Decision Support for z/OS tables and views.</p> <p>This required field can be up to 8 alphanumeric characters. The first character must be alphabetic.</p> <p>The default is DRLUSER. If the value is something other than DRLUSER, it was changed during installation.</p> <p>You can change this value to your user ID if you have DB2 authorization to Tivoli Decision Support for z/OS tables and views.</p>			

Dialog parameters - variables and fields

<i>userid.DRLFPROF</i> variable name	Dialog Parameters field name	Default value	Your value
def_qmflng	QMF language	PROMPTED	
<p>The QMF language for creating reports and queries, either SQL (structured query language) or PROMPTED QUERY.</p> <p>PROMPTED QUERY is the default QMF language for Tivoli Decision Support for z/OS.</p> <p>This is a required field, if your installation uses QMF.</p>			
def_qmfprt	SYSOUT class (in QMF)	Q	
<p>The SYSOUT class for report data sets that QMF generates, or for output that QMF routes to a printer. The default value is Q.</p> <p>This is a required field, if your installation uses QMF.</p>			
def_printer	Default printer	blank	
<p>The GDDM nickname of a printer to use for printing graphic reports. The printer should be one capable of printing GDDM-based graphics.</p> <p>The printer name must be defined in the GDDM nicknames file, allocated to the ADMDEFS ddname. Refer to <i>QMF: Reference</i> and <i>GDDM User's Guide</i> for more information about defining GDDM nicknames.</p>			
def_drlprt	Batch print SYSOUT class	A	
<p>This field is used only if your installation does not use QMF.</p> <p>A valid SYSOUT class for printing tabular reports in batch. Valid values are A-Z, 0-9, and *.</p>			
def_pagelen	Printer line count per page	60	
<p>This field is used only if your installation does not use QMF.</p> <p>The number of report lines that should be printed on each page when you print tabular reports online and in batch.</p>			
def_drlmax	SQLMAX value	5000	
<p>The maximum number of rows for any single retrieval from a Tivoli Decision Support for z/OS table when using a Tivoli Decision Support for z/OS-DB2 interface for such functions as listing tables, reports, or log definitions.</p> <p>The value of this required field is the maximum allowed size of the Tivoli Decision Support for z/OS DB2 table to be retrieved. The default value is 5000 rows of data.</p>			
def_rptdiag	Reporting dialog mode	1	
<p>The dialog mode for using the reporting dialog. Any option you save applies to future sessions.</p> <p>You can choose administrator mode to access reports belonging to all users if you have a Tivoli Decision Support for z/OS administrator authority. You can choose end user mode to access reports that you have created or that have been created for you (including public reports).</p> <p>Type 1 to use end user mode or 2 to specify administrator mode. If you leave the field blank, the default is end user mode.</p>			
N/A	Dialog language	1	
<p>The language in which Tivoli Decision Support for z/OS displays all its windows.</p> <p>Tivoli Decision Support for z/OS supports those languages listed in the window. Choose the language your site has installed.</p> <p>If you leave this field blank, Tivoli Decision Support for z/OS displays its windows in English.</p> <p>Any changes you make to this field become effective in your next dialog session, when Tivoli Decision Support for z/OS allocates its libraries.</p>			
def_db2dspfx	DB2 data sets-prefix	DSN710	

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
<p>The prefix to which Tivoli Decision Support for z/OS appends DB2 data set names as it performs tasks.</p> <p>This field is required if db2def is SUFFIX. If db2def is DATASET, this field is ignored.</p> <p>This field can be 35 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default is DB2.V810. If the value of this field is something other than DB2.V810, it was changed during installation.</p> <p>Any changes you make to this field become effective in your next session, when Tivoli Decision Support for z/OS allocates DB2 libraries and data sets.</p>			
DB2 data sets-suffix	def_db2dssfx	blank	
<p>The suffix that Tivoli Decision Support for z/OS appends as the low-level qualifier for DB2 data sets that Tivoli Decision Support for z/OS uses. Most sites do not use a DB2 data set suffix, but this depends on your DB2 naming conventions.</p> <p>This field can be used if db2def is SUFFIX. If db2def is DATASET, this field is ignored.</p> <p>This field can be 35 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>Your Tivoli Decision Support for z/OS administrator can set a default value for this field if it is in use at your site. If the field is blank, it is very likely not in use.</p> <p>Do not use this field to qualify data sets that you create; this is not its purpose. Use it to identify DB2 modules only.</p> <p>Any changes you make to this field are not effective until your next invocation of the dialog, when Tivoli Decision Support for z/OS has a chance to reallocate DB2 libraries and data.</p>			
def_qmfdspfx	QMF data sets prefix	QMF710	
<p>This field is used only if your installation uses QMF. The prefix to which Tivoli Decision Support for z/OS appends all QMF data set names. This includes all QMF libraries allocated by the dialog during invocation. It also includes all QMF queries and forms.</p> <p>If qmfdef is SUFFIX, this field is required. If qmfdef is DATASET, this field is ignored.</p> <p>This field can be up to 35 alphanumeric characters. Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default is DB2.V810. If the value is something other than DB2.V810, it was changed during installation.</p> <p>Do not use this value to identify your personal QMF data sets. Tivoli Decision Support for z/OS uses this value for all QMF data sets.</p> <p>Any changes you make to this field become effective in your next session, when Tivoli Decision Support for z/OS allocates its libraries.</p>			
def_dsnpref	Tivoli Decision Support for z/OS data sets prefix	DRL182	
<p>The prefix of Tivoli Decision Support for z/OS libraries.</p> <p>This required field can be up to 35 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default is DRL182. If the value of this field is something other than DRL182, it was changed during installation.</p> <p>Any changes you make to this field become effective in your next session, when Tivoli Decision Support for z/OS allocates its libraries.</p>			
No equivalent	Temporary data sets prefix	<i>user_ID</i>	

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
			<p>The prefix for any temporary data sets you create while using Tivoli Decision Support for z/OS.</p> <p>This required field can be up to 35 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default value is your user_ID or the TSO_prefix.user_ID.</p>
def_dsnlocdn	Local definitions data set	DRL.LOCAL.DEFS	
			<p>The partitioned data set (PDS) that contains definitions of Tivoli Decision Support for z/OS objects you have created. The value of this field depends on naming conventions that apply to Tivoli Decision Support for z/OS.</p> <p>The members of this PDS contain definition statements that define new objects to Tivoli Decision Support for z/OS. Tivoli Decision Support for z/OS uses the value of this field to locate local definition members.</p> <p>This optional field can be 44 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default PDS is DRL.LOCAL.DEFS. Your administrator can set a different default for this field during installation. Do not change the value that your Tivoli Decision Support for z/OS administrator sets.</p> <p>Any changes you make to this field are not effective until you start the dialog again, when Tivoli Decision Support for z/OS reallocates local definition data sets.</p>
def_usrlocdn	Local User alter/definitions data set	DRL.LOCAL.USER.DEFS	
			<p>The partitioned data set (PDS) that contains definitions of Tivoli Decision Support for z/OS objects you have modified. The value of this field depends on naming conventions that apply to Tivoli Decision Support for z/OS.</p> <p>The members of this PDS contain definition statements that define user modified objects to Tivoli Decision Support for z/OS. This PDS also contains members with alter statements built by the update processor on the definitions contained in the same PDS. Tivoli Decision Support for z/OS uses the value of this field to locate local user definition members.</p> <p>This optional field can be 44 alphanumeric characters. Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default PDS is DRL.LOCAL.USER.DEFS. Your administrator can set a different default for this field during installation. Do not change the value that your Tivoli Decision Support for z/OS administrator sets.</p> <p>Any changes you make to this field are not effective until you start the dialog again, when Tivoli Decision Support for z/OS reallocates local definition data sets.</p>
def_modform	The local GDDM formats data set	DRL.LOCAL.ADMCFORM	
			<p>The data set where you keep your GDDM formats for graphic reports.</p>
def_drlmsgs	Local messages data set	DRL.LOCAL.MESSAGES	
			<p>Use this field to identify a PDS that contains messages generated by users during communication with Tivoli Decision Support for z/OS administrators.</p> <p>The value of this field depends on naming conventions that your Tivoli Decision Support for z/OS administrator has established.</p> <p>This required field can be up to 44 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>Any changes you make to this field are not effective until you start the dialog again, when Tivoli Decision Support for z/OS reallocates the message data set.</p>
def_dsnreprt	Saved reports data set	DRL.LOCAL.REPORTS	

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
<p>The PDS where Tivoli Decision Support for z/OS saves your tabular reports.</p> <p>This optional field can be up to 44 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default PDS is DRL.LOCAL.REPORTS.</p>			
def_dsncrtrs	Saved charts data set	DRL.LOCAL.CHARTS	
<p>The PDS where Tivoli Decision Support for z/OS saves the graphic reports you choose to save.</p> <p>This optional field can be up to 44 alphanumeric characters.</p> <p>Names longer than 8 characters must be in groups of not more than 8 characters, separated by periods. The first character of each group must be alphabetic.</p> <p>The default PDS is DRL.LOCAL.ADMGDF.</p>			
def_jclsta1, def_jclsta2, def_jclsta3, def_jclsta4	Job statement information (required for batch jobs)	Sample job card in which Tivoli Decision Support for z/OS dynamically substitutes the user ID.	
<p>The job statement information to be used for batch jobs that the dialogs create for you.</p> <p>You must use correct JCL in the job statement. Tivoli Decision Support for z/OS does not validate job statement information.</p> <p>Do not use JCL comments in these JCL statements.</p> <p>You can specify up to four card images in these job statement fields.</p> <p>The first "/" card image should contain the job name. Press Enter to save any job statements for all future sessions.</p>			
bkmgr_mlib	N/A	EOY.SEOYMENU	
The BookManager message library.			
bkmgr_plib	N/A	EOY.SEOYPENU	
The BookManager panel library.			
bkmgr_tlib	N/A	EOY.SEOYTENU	
The BookManager tables library.			
drl1sh00	N/A	IBMBK.DRL1SH00. BKSHELF	
<p>The Tivoli Decision Support for z/OS BookManager bookshelf that contains the names of Tivoli Decision Support for z/OS online books. Change this to reflect the name used in your installation. For example, you might have to increment the 00 in DRL1SH00 because you have installed a newer version of the online books and bookshelf.</p> <p>If you do not plan to use BookManager, change the value of this field to blank.</p>			
dsnsufx	N/A	SDRLDEFS	
The Tivoli Decision Support for z/OS definitions data set suffix.			
execsfx	N/A	SDRLEXEC	
The Tivoli Decision Support for z/OS exec data set suffix.			
loadsfx	N/A	SDRLLOAD	
The Tivoli Decision Support for z/OS load library suffix.			
skelsfx	N/A	SDRLSKEL	
The Tivoli Decision Support for z/OS skeleton data set suffix.			

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
eng_lib_sfx	N/A	ENU	
The English library suffix.			
jpn_lib_sfx	N/A	JPN	
The Japanese library suffix.			
def_nlslang	N/A	eng_lib_sfx	
The national language library suffix.			
repsufx	N/A	"SDRLR"+def_nlslang	
The Tivoli Decision Support for z/OS report definitions library suffix.			
plibsfx	N/A	"SDRLP"+def_nlslang	
The Tivoli Decision Support for z/OS panel library suffix.			
messsfx	N/A	"SDRLM"+def_nlslang	
The Tivoli Decision Support for z/OS message library suffix.			
formsfx	N/A	"SDRLF"+def_nlslang	
The Tivoli Decision Support for z/OS GDDM formats library suffix.			
eng_qmf_sfx	N/A	E	
The English library suffix.			
jpn_qmf_sfx	N/A	K	
The Kanji-Japanese library suffix.			
def_qmflang	N/A	eng_qmf_sfx	
The national language default library suffix.			
qmfdef	N/A	SUFFIX	
<p>The method of describing QMF library names to Tivoli Decision Support for z/OS, either SUFFIX or DATASET.</p> <p>If qmfdef is SUFFIX (the default), Tivoli Decision Support for z/OS implements the QMF library naming standard, requiring a prefix for QMF data sets (def_qmfdspfx) and a suffix (described below). Tivoli Decision Support for z/OS appends each suffix to the QMF prefix to identify QMF libraries, which it then allocates.</p> <p>If qmfdef is DATASET, Tivoli Decision Support for z/OS does not use a prefix or suffix and you must specify fully-qualified data set names for the QMF library variables described below.</p> <p>In either case, Tivoli Decision Support for z/OS uses the next several variables to allocate QMF libraries.</p>			
qmfclib	N/A	SDSQCLST+def_qmflang	
The QMF CLIST library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfclibe	N/A	SDSQCLST+eng_qmf_sfx	
The English QMF CLIST library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx. Tivoli Decision Support for z/OS requires this library even though you might be using another language.			
qmfelib	N/A	SDSQEXEC+def_qmflang	
The QMF EXEC library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfelibe	N/A	SDSQEXEC+eng_qmf_sfx	
The English QMF EXEC library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx. Tivoli Decision Support for z/OS requires this library even though you might be using another language.			

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
qmfplib	N/A	SDSQPLIB+def_qmflang	
The QMF panel library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfmlib	N/A	SDSQMLIB+def_qmflang	
The QMF message library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfslib	N/A	SDSQSLIB+def_qmflang	
The QMF skeleton library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfmap	N/A	SDSQMAP+def_qmflang	
The ADMGGMAP library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmf.pnl	N/A	DSQPNL+def_qmflang	
The QMF panel library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
dsq.pnl	N/A	DSQPNL+def_qmflang	
The ddname of QMF DSQPNLx library. Even if you use fully-qualified data set names to identify QMF data sets, you must specify the ddname of your DSQPNLx library as the value of this variable.			
qmfload	N/A	SDSQLOAD	
The QMF load library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfchart	N/A	DSQCHART	
The ADMCFORM library, which (depending on the value of qmfdef), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_qmfdspfx.			
qmfdsdum	N/A	DUMMY	
The fully-qualified name of the data set to be allocated to ddname DSQDUMP, or DUMMY.			
qmfdebug	N/A	DUMMY	
The fully-qualified name of the data set to be allocated to ddname DSQDEBUG, or DUMMY.			
dsunit	N/A	SYSDA	
The disk unit.			
db2ver	N/A	10	
The version of DB2. Must be a decimal number 1-99.			
db2rel	N/A	1	
The release of DB2.			
db2def	N/A	SUFFIX	

Dialog parameters - variables and fields

<i>userid</i> .DRLFPROF variable name	Dialog Parameters field name	Default value	Your value
<p>The method of describing DB2 library names to Tivoli Decision Support for z/OS, either SUFFIX or DATASET.</p> <p>If db2def is SUFFIX (the default), Tivoli Decision Support for z/OS implements the DB2 library naming standard, requiring a prefix for DB2 data sets (def_db2dspfx), a library name, and an optional suffix (def_db2dssfx).</p> <p>If db2def is DATASET, Tivoli Decision Support for z/OS does not use a prefix or a suffix and you must specify fully-qualified data set names for the DB2 library variables described below.</p> <p>In either case, Tivoli Decision Support for z/OS uses the next several variables to allocate DB2 libraries.</p>			
db2llib	N/A	RUNLIB.LOAD	
<p>The DB2 runlib load library name, which (depending on the value of db2def), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.</p>			
db2load	N/A	SDSNLOAD	
<p>The DB2 load library, which (depending on the value of db2def), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.</p>			
db2clst	N/A	SDSNCLIST	
<p>The DB2 CLIST library, which (depending on the value of db2def), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.</p>			
db2mlib	N/A	SDSNSPFM	
<p>The DB2 message library, which (depending on the value of db2def), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.</p>			
db2plib	N/A	SDSNSPFP	
<p>The DB2 panel library, which (depending on the value of db2def), is the fully-qualified name or is a value that Tivoli Decision Support for z/OS appends to def_db2dspfx before appending def_db2dssfx.</p>			
gddmload	N/A	GDDM.SADMMOD	
<p>The GDDM load library.</p>			
admsymb1	N/A	GDDM.SADMSYM	
<p>The GDDM symbols library.</p>			
admdefs	N/A	SYS1.GDDMNICK	
<p>The GDDM nicknames library.</p>			
admprntq	N/A	None	
<p>The data set name of the GDDM master print queue, if any. This overrides any value specified for TSOPRNT in the GDDM external defaults file. If you supply a value, Tivoli Decision Support for z/OS adds an ADMPRNTQ DD statement to the batch JCL for graphic reports.</p>			
def_geapplid	N/A	zuser	
<p>The application ID (usually sent as a TSO user ID) that has an assigned Information/Management privilege class. The default is the user ID of the Tivoli Decision Support for z/OS user.</p>			
def_gesessn	N/A	BLGSES00	
<p>The session member (module) used to start an Information/Management session.</p>			
def_geprivcl	N/A	MASTER	
<p>The privilege class specified in an Information/Management group record.</p>			
VIEWER	N/A	NO	

Dialog parameters - variables and fields

<i>userid.DRLFPROF</i> variable name	Dialog Parameters field name	Default value	Your value
Specifies and enables the use of the Viewer. This parameter should be YES, unless you never use the Viewer. If the value here is YES, you can also run your own reports any time in the future in the Viewer application. Any other value than YES or NO causes Tivoli Decision Support for z/OS to use YES.			
qmfuse	N/A	YES	
Specifies if QMF is used with Tivoli Decision Support for z/OS in your installation. Any other value than YES or NO causes Tivoli Decision Support for z/OS to use YES.			
gddmuse	N/A	YES	
Specifies if GDDM is used with Tivoli Decision Support for z/OS in your installation. (If QMF is used, GDDM must be used.) If GDDM is not used, reports are always shown in tabular format. Any other value than YES or NO causes Tivoli Decision Support for z/OS to use YES.			
decsep	N/A	PERIOD	
When generating tabular reports without QMF, Tivoli Decision Support for z/OS uses period as decimal separator and comma as thousands separator. You can exchange the decimal and thousands separators by specifying decsep="COMMA". In that case, period is used as thousands separator. Any other value of decsep causes Tivoli Decision Support for z/OS to use period as a decimal separator.			
subhdrv	N/A	N	
This value is used only for QMF (where qmfuse='YES'). Specify Y if you want Tivoli Decision Support for z/OS to replace empty variables in the report header with a text string. You specify the text string using F11 on the Data Selection panel, or when you get message DRLA171. Note: Replacing empty variables increases the time taken to generate a report. Specify N to leave the empty variable in the report.			
def_useaot	N/A	NO	
Specifies whether Analytics component tables are created as Accelerator Only Tables in IBM DB2 Analytics Accelerator or as tables in DB2. "YES": Tables are created as Accelerator Only Tables. "NO": Tables are created in DB2 and are applicable for use either as DB2 tables or as IDAA_ONLY table. The default value is "NO". This parameter is only applicable for Analytics components.			
def_accelerator	N/A		
The name of the Accelerator where the Analytics components tables reside. Required only if using Accelerator Only Tables, that is, if <i>def_useaot</i> is set to "YES". This parameter is only applicable for Analytics components.			
def_timeint	N/A	T	
Specifies the time interval granularity for records collected for Analytics components tables. "H": The timestamp for records is rounded to hourly intervals, which is similar to non-Analytics tables with a suffix of "_H" in other components. "S": The timestamp for records is rounded to intervals of a second, which is similar to non-Analytics tables with time field instead of timestamp in other components. "T": The timestamp for tables is the actual timestamp in the SMF log record, which is similar to non-Analytics tables with suffix "_T". The default value is "T". This parameter is only applicable for Analytics components.			

Allocation overview

This section describes the data sets allocated by Tivoli Decision Support for z/OS.

Library type or data set ddname	Library or data set	Allocated by (EPDM exec)
Tivoli Decision Support for z/OS allocates the following libraries as a user starts a Tivoli Decision Support for z/OS dialog:		
ISPPLIB	<ul style="list-style-type: none"> Tivoli Decision Support for z/OS panel library QMF panel library DB2 panel library 	DRLEINI1
ISPTLIB	<ul style="list-style-type: none"> Tivoli Decision Support for z/OS tables library QMF tables library BookManager tables library 	DRLEINI1
ISPMLIB	<ul style="list-style-type: none"> Tivoli Decision Support for z/OS message library QMF message library DB2 message library 	DRLEINI1
ISPLLIB	<ul style="list-style-type: none"> Tivoli Decision Support for z/OS load library QMF load library 	DRLEINI1
ISPSLIB	<ul style="list-style-type: none"> Tivoli Decision Support for z/OS skeleton library QMF skeleton library 	DRLEINI1
Tivoli Decision Support for z/OS allocates the following data sets as a user starts a Tivoli Decision Support for z/OS dialog:		
DRLTABL	Userprefix.DRLTABL (for values in query variables)	DRLEINI1
ADMGDF	Saved charts data set	DRLEINI1
DRLMSGDD	Tivoli Decision Support for z/OS user message data set (drlmsgs)	DRLEINI1
Tivoli Decision Support for z/OS allocates the following libraries as a user starts a Tivoli Decision Support for z/OS function that uses QMF:		
SYSPROC	QMF CLIST library (def_qmfdspfx.qmfclib+E)	DRLEQMF
SYSEXEC	QMF exec library (def_qmfdspfx.qmfelib+E)	DRLEQMF
ADMGGMAP	SDSQMAP library (def_qmfdspfx.qmfmap)	DRLEQMF
ADMCFORM	Saved forms data set + DSQCHART library (dsnpref.formsfx + def_qmfdspfx.qmfchart)	DRLEQMF
DSQUCFRM	Saved forms data set	DRLEQMF
DSQPNLE	QMF panel library	DRLEQMF
DSQPRINT	QMF sysout class (qmfprt)	DRLEQMF
DSQPILL	NEW DELETE (temporary file allocation)	DRLEQMF
DSQEDIT	NEW DELETE (temporary file allocation)	DRLEQMF
DSQDEBUG	(qmfdebug)	DRLEQMF
DSQUDUMP	(qmfdsdum)	DRLEQMF
Tivoli Decision Support for z/OS allocates the following library as a user starts a Tivoli Decision Support for z/OS function that uses GDDM:		
ADMSYMBL	GDDM symbols data set	DRLEINI1
Tivoli Decision Support for z/OS allocates the following libraries when a table or report is displayed without QMF:		
DRLTAB	Userprefix.DRLTAB (for table display)	DRLEADIT
DRLREP	Userprefix.DRLREP (for report display)	DRLERDIR

Allocation overview

Library type or data set ddname	Library or data set	Allocated by (EPDM exec)
Tivoli Decision Support for z/OS allocates the following library as a user starts DB2 Interactive (DB2I) from Tivoli Decision Support for z/OS:		
SYSPROC	DB2 CLIST library (db2dspfx.db2clst)	DRLEDB2I

Chapter 4. Overview of Tivoli Decision Support for z/OS objects

This section describes how a feature definition member is used to update system tables. It then describes how Tivoli Decision Support for z/OS uses the resulting component definitions to install a component's objects. There is also a section on how to create and change definitions in both the dialog and log collector language.

For more information about the log collector language and report definition language statements, see the *Language Guide and Reference*, SH19-6817.

This topic uses the Sample component as the basis of most of its examples. For more information, see Chapter 13, "Sample components," on page 253.

For information on the naming convention for Tivoli Decision Support for z/OS definition members, see Chapter 5, "Naming convention for Tivoli Decision Support for z/OS definition members," on page 81.

How Tivoli Decision Support for z/OS component installation works

Component installation starts with the SMP/E installation of the definition members of a feature in the DRL181.SDRLDEFS library. Tivoli Decision Support for z/OS features provide definition members that update the system tables with information about the definitions in a feature.

Defining definition library members with SQL

Before installing TDS for z/OS components, you must create or update the system tables. When you do this from the dialog or in batch, the DRLxxxx members, in the DRL182.SDRLDEFS library, contain SQL statements that are executed.

Figure 17 on page 72 shows the DRLxxxx definition member for the Sample component. These members use the SQL log collector language statement to pass an SQL statement to DB2.

```

/*****/
/* Sample Component */
/*****/
SQL INSERT INTO &SYSPREFIX.DRLCOMPONENTS
  (COMPONENT_NAME, DESCRIPTION, USER_ID)
  VALUES('SAMPLE','Sample Component',USER);
/*****/
/* Log and record definitions */
/*****/
SQL INSERT INTO &SYSPREFIX.DRLCOMP_OBJECTS
  (COMPONENT_NAME, OBJECT_TYPE, OBJECT_NAME, MEMBER_NAME)
  VALUES('SAMPLE','LOG ','SAMPLE','DRLLSAMP');
:
/*****/
/* Tablespace, table, and update definitions */
/*****/
SQL INSERT INTO &SYSPREFIX.DRLCOMP_OBJECTS
  (COMPONENT_NAME, OBJECT_TYPE, OBJECT_NAME, MEMBER_NAME)
  VALUES('SAMPLE','TABSPACE','DRLSSAMP','DRLSSAMP');
:
/*****/
/* Report and report group definitions */
/*****/
SQL INSERT INTO &SYSPREFIX.DRLCOMP_OBJECTS
  (COMPONENT_NAME, OBJECT_TYPE, OBJECT_NAME, MEMBER_NAME)
  VALUES('SAMPLE','REPGROUP','SAMPLE','DRLOSAMP');
:

```

Figure 17. Tivoli Decision Support for z/OS definition member DRLISAMP, setting component definitions

Executing these statements populates the Tivoli Decision Support for z/OS system tables with component definitions. These component definitions describe the installable components and the SDRLDEFS members that can be used to install the component.

How Tivoli Decision Support for z/OS controls object replacement

Once the system tables have been updated with the installation members, you must reinstall all affected components in order to replace all objects. Each component installed is controlled by a variable VERSION which is specified in the DEFINE statements and a corresponding column VERSION is included in the Tivoli Decision Support for z/OS system tables where objects are defined.

During the installation of the Tivoli Decision Support for z/OS components, a preprocessor checks each definition member to see if an object already exists (from the installation of an earlier level of the component).

If the object *does not* already exist, the DEFINE statement for this object is passed to the log collector.

If the object *does* already exist, and providing the variable VERSION is specified in the DEFINE statement for the object, then the values of VERSION in the DEFINE statement and in the system table where the object is defined, are compared. If the values of VERSION are the same, the log collector replaces the DEFINE statement for the object with a comment, saying that the most recent version of the object already exists in the system table. If the values of VERSION are different, the log collector inserts a DROP statement. This DROP statement drops the object so that it can be redefined.

Note: Tivoli Decision Support for z/OS only checks the VERSION variable when you install using option 2 Components.

All log, record, record procedure, and update objects shipped with the product contain the VERSION variable, which takes the value:

IBM.xxx

where xxx corresponds to the product version. For example, IBM.182 indicates objects created or modified by Tivoli Decision Support for z/OS 1.8.2. If an object is modified by an APAR, then the APAR number is used as the VERSION variable, for example, VERSION 'PK28980'.

Tivoli Decision Support for z/OS Version variable format

All Tivoli Decision Support for z/OS log, record, record procedure, and update objects shipped with the product contain the VERSION variable, which takes the value IBM.xxx, where xxx corresponds to the product version. For example IBM.181 indicates objects created or modified by Tivoli Decision Support for z/OS 1.8.1.

If an object is modified by an APAR, then the APAR number is used as the VERSION variable, for example, VERSION 'PK28980.xxx'.

Tivoli Decision Support for z/OS recognizes the following version variable patterns as being standard objects shipped by the product:

- Version numbers beginning with 'IBM'.
- Version numbers with no text (the empty string or no version clause).
- Version numbers beginning with an APAR number, that is, two letters followed by any number of digits up to an optional decimal point. For example, the version numbers PM123, PX123456.V181, RW987654, and OK123.2014101, are all considered 'standard' version numbers, but PK1234A and MXC1234 are not.

Custom Version numbers:

When customizing Tivoli Decision Support for z/OS objects (see “Controlling objects that you have modified” on page 136) you must choose a version number which does not conform to the standards above. A version number might be 'ALTERED' or 'MODIFIED', or your own version system such as 'C.V2'.

How Tivoli Decision Support for z/OS determines installation order

After Tivoli Decision Support for z/OS stores the names of a feature's component objects and definition members in the system tables, you can use the dialog to install the feature's components. The product queries the system tables to determine the names of definition members in the DRL182.SDRLDEFS, DRL182.SDRLRxxx, and DRL182.SDRLFxxx libraries. (xxx is ENU for the English language version of Tivoli Decision Support for z/OS and JPN for the Japanese version.

Tivoli Decision Support for z/OS requires some definitions to exist before it can install others. For example, if a component contains a record procedure, The product must install the record definition that maps the source record for the record procedure before installing the record procedure. Furthermore, it must install the record procedure before installing the record definition that maps the

Defining objects

record procedure's output. The definition members that are supplied by the product supplies often combine several definitions in the same member to ensure their order of installation.

Table 5 shows the order in which Tivoli Decision Support for z/OS installs a feature's definitions.

Table 5. Order of installation of feature definition members

Order	Member naming convention	Definition types
1	DRLxxxx	Logs.
2	DRLRxxxx	Records and record procedures. Record definitions mapping record procedure input must appear before the associated record procedure definition. Record definitions mapping record procedure output must appear after the associated record procedure definition.
3	DRLSxxxx	Tablespaces.
4	DRLTxxxx	Lookup tables, tables, updates, and views. Lookup tables and tables must be defined before update definitions that refer to them. Tables must also be defined before views that refer to them.
5	DRLUxxxx	Updates (also found in DRLTxxxx members).
6	DRLVxxxx	Views (also found in DRLTxxxx members).
7	DRLOxxxx	Report groups and reports. Report groups must be defined before the report definitions that reference them.

The order of installation within a definition type is determined by the sort sequence of the definition member names. The examples that follow appear in the same order that Tivoli Decision Support for z/OS would install them.

Defining logs with log collector language

DRLxxxx members of the DRL182.SDRLDEFS library define log types to Tivoli Decision Support for z/OS. Figure 18 shows the definition member for the SAMPLE log type.

```
DEFINE LOG SAMPLE VERSION 'IBM.110';  
COMMENT ON LOG SAMPLE IS 'Sample log definition';
```

Figure 18. Tivoli Decision Support for z/OS definition member DRLLSAMP, defining a log type

Defining records with log collector language

DRLRxxxx members of the DRL182.SDRLDEFS library define record types to Tivoli Decision Support for z/OS. Figure 19 on page 75 shows the definition for the SAMPLE_01 record type. (Chapter 14, "Record definitions supplied with Tivoli Decision Support for z/OS," on page 259 describes Tivoli Decision Support for z/OS record definitions.)

```

DEFINE RECORD SAMPLE_01
  VERSION 'IBM.110'
  IN LOG SAMPLE
  IDENTIFIED BY S01TYPE = '01'
  FIELDS
    (S01TYPE OFFSET 4 LENGTH 2 CHAR,
     S01DATE OFFSET 7 DATE(MMDDYY),
     S01TIME OFFSET 14 TIME(HHMMSS),
     S01SYST OFFSET 21 LENGTH 4 CHAR,
     S01USER OFFSET 26 LENGTH 8 CHAR,
     S01TRNS OFFSET 35 LENGTH 6 EXTERNAL INTEGER,
     S01RESP OFFSET 42 LENGTH 6 EXTERNAL INTEGER,
     S01CPU OFFSET 49 LENGTH 6 EXTERNAL INTEGER,
     S01PRNT OFFSET 56 LENGTH 6 EXTERNAL INTEGER);

COMMENT ON RECORD SAMPLE_01 IS 'Sample record type 01';

```

Figure 19. Tivoli Decision Support for z/OS definition member DRLRSAMP, defining a record type

Defining table spaces

DRLSxxxx members of the DRL182.SDRLDEFS library define table spaces to Tivoli Decision Support for z/OS. The product defines at least one table space per component to contain all the tables in the component. In many cases there is one table space per table. Table spaces can be defined explicitly using SQL statements. Alternatively, the Log Collector Language statement GENERATE TABLESPACE may be used to create table spaces, which use values in the GENERATE_PROFILES and GENERATE_KEYS system tables to determine the partitioning type. Using the GENERATE statement eliminates the need to change the DRLSxxxx members, and allows multiple table spaces to be configured in the manner. The following figure shows the definition for the DRLSSAMP table space of the Sample component.

```

SQL CREATE TABLESPACE DRLSSAMP
  IN &DATABASE
  USING STOGROUP &STOGROUP
    PRIQTY 60
    SECQTY 30
  SEGSIZE 8
  BUFFERPOOL &TSBUFFERPOOL
  LOCKSIZE TABLE;

```

Figure 20. Tivoli Decision Support for z/OS definition member DRLSSAMP, defining a table space

Defining tables and updates

DRLTxxxx members of the DRL182.SDRLDEFS library define tables and updates to Tivoli Decision Support for z/OS. These members use the SQL log collector language statement to create tables in the database, populate lookup tables, and grant access to the tables. They also use the DEFINE UPDATE log collector language statement to create update definitions in the system tables. To give an example, Figure 21 on page 76 and Figure 22 on page 77 show the definition for tables (that includes the lookup table) and updates of the Sample component, DRLTSAMP. Figure 21 on page 76 uses the SQL log collector language statement and Figure 22 on page 77 uses the DEFINE UPDATE log collector language statement.

Defining objects

```

/*****
/* Define table SAMPLE_USER
/*****
SQL CREATE TABLE &PREFIX.SAMPLE_USER
  (USER_ID          CHAR(8) NOT NULL,
   DEPARTMENT_NAME CHAR(8) NOT NULL,

   PRIMARY KEY (USER_ID))
  IN &DATABASE.DRLSSAMP;

SQL CREATE UNIQUE INDEX &PREFIX.SAMPUSER_IX
  ON &PREFIX.SAMPLE_USER
  (USER_ID)
  USING STOGROUP &STOGROUP.
  PRIQTY 12
  SECQTY 4
  CLUSTER
  BUFFERPOOL &IXBUFFERPOOL;
/*****
/* Define comments for SAMPLE_USER
/*****
SQL COMMENT ON TABLE &PREFIX.SAMPLE_USER
  IS 'This lookup table assigns department names to users.';

SQL COMMENT ON &PREFIX.SAMPLE_USER
  (USER_ID          IS 'User ID.',
   DEPARTMENT_NAME IS 'Department name.');
```

```

/*****
/* Grant users read access to SAMPLE_USER
/*****
SQL GRANT SELECT ON &PREFIX.SAMPLE_USER TO &USERS.;
/*****
/* Insert data in SAMPLE_USER
/*****
SQL INSERT INTO &PREFIX.SAMPLE_USER
  VALUES('ADAMS  ', 'App1 Dev');
:
:
/*****
/* Define table SAMPLE_H
/*****
SQL CREATE TABLE &PREFIX.SAMPLE_H
  (DATE          DATE NOT NULL,
   TIME         TIME NOT NULL,
   SYSTEM_ID    CHAR(4) NOT NULL,
   DEPARTMENT_NAME CHAR(8) NOT NULL,
   USER_ID     CHAR(8) NOT NULL,
   TRANSACTIONS INTEGER,
   RESPONSE_SECONDS INTEGER,
   CPU_SECONDS  FLOAT,
   PAGES_PRINTED INTEGER,
   PRIMARY KEY (DATE, TIME, SYSTEM_ID, DEPARTMENT_NAME, USER_ID))
  IN &DATABASE.DRLSSAMP;
```

Figure 21. Tivoli Decision Support for z/OS definition member DRLTSAMP, defining tables and updates (Part 1 of 2)

```

:
/* ***** */
/* Define update from record SAMPLE_01 */
/* ***** */
DEFINE UPDATE SAMPLE_01_H
  VERSION 'IBM.110'
  FROM SAMPLE_01
  TO &PREFIX.SAMPLE_H
  GROUP BY
    (DATE          = S01DATE,
     TIME          = ROUND(S01TIME,1 HOUR),
     SYSTEM_ID     = S01SYST,
     DEPARTMENT_NAME = VALUE(LOOKUP DEPARTMENT_NAME
                           IN &PREFIX.SAMPLE_USER
                           WHERE S01USER = USER_ID,
                           '?'),
     USER_ID       = S01USER)
  SET
    (TRANSACTIONS = SUM(S01TRNS),
     RESPONSE_SECONDS = SUM(S01RESP),
     CPU_SECONDS   = SUM(S01CPU/100.0),
     PAGES_PRINTED = SUM(S01PRNT));
:
/* ***** */
/* Define update from SAMPLE_H */
/* ***** */
DEFINE UPDATE SAMPLE_H_M
  VERSION 'IBM.110'
  FROM &PREFIX.SAMPLE_H
  TO &PREFIX.SAMPLE_M
  GROUP BY
    (DATE          = SUBSTR(CHAR(DATE),1,8) || '01',
     SYSTEM_ID     = SYSTEM_ID,
     DEPARTMENT_NAME = DEPARTMENT_NAME,
     USER_ID       = USER_ID)
  SET
    (TRANSACTIONS = SUM(TRANSACTIONS),
     RESPONSE_SECONDS = SUM(RESPONSE_SECONDS),
     CPU_SECONDS   = SUM(CPU_SECONDS),
     PAGES_PRINTED = SUM(PAGES_PRINTED));

```

Figure 22. Tivoli Decision Support for z/OS definition member DRLTSAMP, defining tables and updates (Part 2 of 2)

Defining updates and views

DRLUxxxx members of the DRL182.SDRLEDFS library define updates not previously defined in DRLTxxxx definition members. For example, member DRLUMVAV in the DRL182.SDRLEDFS library defines updates from record types SMF_030 and SMF_070 to the AVAILABILITY_T table.

DRLVxxxx members of the DRL182.SDRLEDFS library define views not previously defined in DRLTxxxx definition members. For example, member DRLVC901 in the DRL182.SDRLEDFS library defines views on the CICS_T_TRAN_T table for CICS unit-of-work processing.

Defining table spaces and indexes using the GENERATE statement

The GENERATE statement may be used to create table spaces, partitioning on tables, and indexes on tables. The GENERATE statement has a PROFILE parameter which is the major key to the GENERATE_PROFILES and GENERATE_KEYS

Defining objects

| system tables. All customization for creating table spaces, partitioning, and indexes
| may be performed using these system tables. The definition member DRLTKEYS is
| used to create and load the default values into the GENERATE_PROFILES and
| GENERATE_KEYS when the system tables are created.

| These system tables provide default profiles for IMS, SMF, and the SMF CICS
| partitioned components. The supplied defaults may be changed by updating the
| data in these system tables without modifying the GENERATE statements in the
| Tivoli Decision Support for z/OS definition members. The profiles may be made
| more granular by using the COMPONENT_ID, SUBCOMPONENT_ID or
| TABLESPACE_NAME key fields, with no changes required to the product
| definition members.

| For Tivoli Decision Support for z/OS 1.8.2, the use of the GENERATE statement is
| implemented for all Key Performance Metrics components and all CICS Partitioned
| components.

| When you select a table space type of RANGE to create a Range Partitioned
| Universal Table space, the number of partitions created is determined by the
| number of entries for the key in the GENERATE_KEYS system table. For example
| the supplied profile of IMS has PART_NUM 1-4 which will generate 4 partitions
| for a RANGE table. This may be increased or decreased to generate the number of
| partitions you require. Changing the number of partitions does not require a
| change to the GENERATE statement in the Tivoli Decision Support for z/OS
| definition members.

Defining reports

DRLOxxx members of the DRL182.SDRLRENU library use report definition language to define report groups and reports in Tivoli Decision Support for z/OS system tables. Report definition members are contained in national language-specific definition libraries.

Figure 23 on page 79 shows the definition for the reports and report group of the Sample component.

```

DEFINE GROUP SAMPLE
  VERSION 'IBM.110'
  DESC 'Sample Reports';

DEFINE REPORT SAMPLE01
  VERSION 'IBM.110'
  DESC 'Sample Report 1'
  QUERY DRLQSA01
  FORM DRLFSA01
  CHART DRLGSURF
  ATTRIBUTES SAMPLE
  GROUPS SAMPLE;

DEFINE REPORT SAMPLE02
  VERSION 'IBM.110'
  DESC 'Sample Report 2'
  QUERY DRLQSA02
  FORM DRLFSA02
  ATTRIBUTES SAMPLE
  GROUPS SAMPLE;

DEFINE REPORT SAMPLE03
  VERSION 'IBM.110'
  DESC 'Sample Report 3'
  QUERY DRLQSA03
  FORM DRLFSA03
  CHART DRLGHORB
  ATTRIBUTES SAMPLE
  GROUPS SAMPLE;

```

Figure 23. Tivoli Decision Support for z/OS definition member DRLOSAMP, defining reports and report groups

The Tivoli Decision Support for z/OS report definition program uses the definitions in DRLOxxxx members to locate these types of members for each report:

Member type

Description

DRLQxxxx

Report queries in DRL182.SDRLRxxx

DRLFxxxx

Report forms in DRL182.SDRLRxxx

DRLGxxxx

Report charts in DRL182.SDRLFxxx

where xxx refers to your national-language code (for example, ENU, JPN).

Tivoli Decision Support for z/OS imports members in these data sets to QMF to provide queries and forms for predefined reports. If QMF is not used, the contents of the report queries and forms are stored in Tivoli Decision Support for z/OS system tables.

DRLQxxxx members in the DRL182.SDRLRENU library are queries for predefined reports. Figure 24 on page 80 shows the query for Sample Report 1.

Defining objects

```
SELECT TIME, DEPARTMENT_NAME, SUM(CPU_SECONDS)
FROM &PREFIX.SAMPLE_H
WHERE SYSTEM_ID = &SYSTEM_ID.
GROUP BY TIME, DEPARTMENT_NAME
```

Figure 24. Tivoli Decision Support for z/OS definition member DRLQSA01, report query

DRLFxxxx members in the DRL182.SDRLRENU library are QMF forms for predefined English reports. For example, DRLFSA01 is the QMF form for Sample Report 1.

DRLGxxxx members in the DRL182.SDRLFENU library are GDDM/ICU formats for predefined English reports. For example, DRLGSURF is the GDDM/ICU format used for Sample Report 1.

Chapter 5. Naming convention for Tivoli Decision Support for z/OS definition members

This section describes the naming convention for members of the DRL182.SDRLDEFS and DRL182.SDRLRENU libraries. For information on defining these libraries, see Chapter 4, “Overview of Tivoli Decision Support for z/OS objects,” on page 71.

Naming convention for members of DRL182.SDRLDEFS

The naming convention for the Tivoli Decision Support for z/OS definitions library is:

Naming convention	Description
--------------------------	--------------------

DRLBxxxx

Log data manager collect statements

DRLIxxxx

Component definitions (SQL statements that are executed when the system tables are created or updated)
--

DRLLxxxx

Log definitions

DRLRxxxx

Record definitions Chapter 14, “Record definitions supplied with Tivoli Decision Support for z/OS,” on page 259 describes record definitions.

DRLSxxxx

Table space definitions

DRLTxxxx

Table and update definitions

DRLUxxxx

Update definitions (when separate from tables)
--

DRLVxxxx

View definitions

DRLWxxxx

Migration definitions

Naming convention for members of DRL182.SDRLRENU

The naming convention for the Tivoli Decision Support for z/OS (predefined) reports definitions library, SDRLRENU (or SDRLRJPN), is:

Naming convention	Description
--------------------------	--------------------

DRLOxxxx

Report definitions

DRLQxxxx

SQL queries

DRLFxxxx

QMF forms

Naming convention for members of DRL181.SDRLRENU

Part 3. Administering Tivoli Decision Support for z/OS

Chapter 6. Setting up operating routines

About this task

This section describes how to develop operating routines for:

- “Collecting log data”
- “Administering the Tivoli Decision Support for z/OS database” on page 95
- “Administering reports” on page 112

The sample jobs described may not be identical to those shipped with Tivoli Decision Support for z/OS. Before using these jobs, refer to the samples in the DRL182.SDRLCNTL library.

Collecting log data

About this task

One of your primary responsibilities is to establish routines to collect data. To do this, you can use either the Tivoli Decision Support for z/OS administration dialog or log collector language statements that you execute through either a job or the dialog. This section describes:

1. How to collect data from the SAMPLE log type. The Sample component contains a log definition, record definitions, and update definitions for collecting SAMPLE log data sets.
2. How to collect data in batch without using the dialog. See “Collecting data from a log into DB2 tables” on page 146 for information about using the dialog to collect data. You can also automate the collection of data using the log data manager option, described in Chapter 10, “Working with the log data manager option,” on page 205.

Collecting data through the administration dialog

About this task

To collect log data from a SAMPLE log data set:

Procedure

1. From the Administration window, select 3, Logs, and press Enter. The Logs window is displayed.
2. From the Logs window, select Sample and press F11. The Collect window is displayed.
3. Type DRL182.SDRLDEFS(DRLSAMPL) in the Data set field. This is the name of the data set that contains log data
4. Press F4 to start an online collect process. After the data collection is complete, Tivoli Decision Support for z/OS displays statistics about the collect. (See “Sample collect messages” on page 90 for more information about the statistics.)
5. When the collect is complete, press F3. The product returns to the Logs window.
6. From the Logs window, press F3. The product returns to the Administration window.

Using log collector language to collect data

About this task

To collect log data using the SAMPLE log definition, create and submit the JCL (Figure 25).

```
//jobname JOB parameters
//LC EXEC PGM=DRLPLC,PARM=('SYSPREFIX=DRLSYS SYSTEM=DSN')
//STEPLIB DD DISP=SHR,DSN=DRLxxx.SDRLLoad
//DRLIN DD *
          COLLECT SAMPLE;
//DRLLLOG DD DISP=SHR,DSN=DRLxxx.SDRLDEFS(DRLSAMPL)
//DRLOUT DD SYSOUT=*
//DRLDUMP DD SYSOUT=*
```

Figure 25. Invoking the log collector in batch to collect data

Tivoli Decision Support for z/OS uses the log collector program (DRLPLC) to collect the SAMPLE log type, using these ddnames:

DD statement name

Description

DRLIN

Contains the log collector language statements. It can contain fixed-length or varying-length records of any length, but the log collector reads a maximum of 72 bytes from each record.

DRLLLOG

Identifies the log data set. The data set attributes are determined by the program creating the log.

DRLOUT

Identifies where collect messages are routed. It can have fixed-length or varying-length records of any length, but the log collector assumes a length of at least 80 bytes for formatting. Lines that are longer than the specified record length are wrapped to the next line. DRLOUT is allocated as RECFM=F and LRECL=80 if no DCB attributes are specified.

DRLDUMP

Identifies where collect diagnostics are routed. It can have fixed-length or varying-length records of any length, but the log collector assumes a length of at least 80 bytes for formatting. Lines that are longer than the specified record length are wrapped to the next line. DRLDUMP is allocated as RECFM=F and LRECL=80 if no DCB attributes are specified.

The DRLJCOLL job

The DRLJCOLL job in the DRL182.SDRLCNTL library is a generic collect job, adaptable for most logs. Figure 26 on page 87 and Figure 27 on page 88 show DRLJCOLL, used to collect data from an SMF log data set.

Note: The log data sets that are used as input for the collect (DRLLLOG DD statement) are expected to be sorted in chronological order.

```

//DRLJCOLL JOB (ACCT#),'COLLECT'
//*****
//* Licensed Materials - Property of IBM *
//* 5698-B06 (C) Copyright IBM Corporation 1993, 2015 *
//* See Copyright Instructions. *
//*****
//* Name: DRLJCOLL *
// *
//* Status: Tivoli Decision Support for z/OS 1.7.1 *
// *
//* Function: *
// * Tivoli Decision Support for z/OS collect job. *
// *
// * Replace "COLLECT SMF" below with one of the following *
// * statements to collect other logs: *
// *
// * COLLECT DCOLLECT *
// * WHERE DCUDATE > DATE(LOOKUP_LAST_DCOLLECT_TIME *
// * IN DRL.DFSMS_LAST_RUN *
// * WHERE DCUSYSID = MVS_SYSTEM_ID *
// * AND DCURCTYP = RECORD_TYPE); *
// * (replace DRL with the table prefix you use) *
// * (the lookup table DFSMS_LAST_RUN must be initialized *
// * before the first collect as described in the DFSMS *
// * customization section of the SP Reference manual) *
// *
// * COLLECT EREP; *
// *
// * SET JES_COMPLEX = ' '; *
// * COLLECT SYSLOG_JES2; *
// *
// * For operations log (OPERLOG) produced using the System *
// * Logger, use the COLLECT statement above and change the *
// * //DRLLLOG statement as follows: *
// * //DRLLLOG DD DSN=SYSPLEX.OPERLOG,DISP=SHR, *
// * DCB=(LRECL=32756, BLKSIZE=32760, RECFM=VB), *
// * SUBSYS=(LOGR,, *
// * 'FROM=(2015/152,00:00),TO=(2015/153,23:59)',) *
// *
// * SET JES_COMPLEX = 'JES3COMP'; *
// * COLLECT SYSLOG_JES3; *
// * (replace JES3COMP with the name of the JES3 complex) *
// *
// * SET MVS_SYSTEM_ID = 'MVS1'; *
// * COLLECT NETVIEW; *
// * (replace MVS1 with the name of the MVS system) *

```

Figure 26. DRLJCOLL job for collecting data from an SMF data set (Part 1 of 2)

Collecting log data

```
/** *
/** COLLECT OPC; *
/** *
/** SET VMID = 'VM1'; *
/** COLLECT VMACCT; *
/** (replace VM1 with the name of the VM system) *
/** *
/** COLLECT VMPRF; *
/** COLLECT VMPERFT; *
/** *
/** COLLECT UNIX; *
/** *
/** COLLECT OS400_JOURNAL; *
/** COLLECT OS400_CONFIG; *
/** COLLECT OS400_HISTORY; *
/** COLLECT OS400_PM_DISK; *
/** COLLECT OS400_PM_POOL; *
/** COLLECT OS400_PM_SYS; *
/** *
/** SET UNLOAD_DATE = 'YYYY-MM-DD'; *
/** SET SYSTEM_ID = 'MVS1'; *
/** COLLECT RACFCONF REPROCESS; *
/** (Replace YYYY-MM-DD with the date when you run the *
/** RACF Database Unload utility. As default, the current *
/** date is used) *
/** (Replace MVS1 with the name of your system. As default, *
/** $UNK is used) *
/** *
/** COLLECT LINUX; *
/** *
/** COLLECT ZLINUX; *
/** *
/** For some logs, special collect jobs are required: *
/** *
/** DRLJCOIM IMS log *
/** DRLJCOVP Network configuration data *
/** DRLJCOIN Tivoli Information Management for z/OS data *
/** *
/** Notes: *
/** Before you submit the job: *
/** - Check the Tivoli Decision Support for z/OS *
/** and DB2 data set names. *
/** - Check the DB2 subsystem name (default is DSN) *
/** and Tivoli Decision Support for z/OS *
/** system table *
/** prefix (default is DRLSYS). *
/** - Insert the correct collect statement in DRLIN *
/** (as described above). *
/** - Specify the name of the log data set in DRLLOG. *
/*******
/**COLLECT EXEC PGM=DRLPLC,PARM=('SYSTEM=DSN SYSPREFIX=DRLSYS')
/**STEPLIB DD DISP=SHR,DSN=DRL181.SDRLLOAD
/** DD DISP=SHR,DSN=DSN710.DSNLOAD
/**DRLIN DD *

COLLECT SMF;

/**DRLLOG DD DISP=SHR,DSN=log-data-set
/**DRLOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
/**DRLDUMP DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)

/*
```

Figure 27. DRLJCOLL job for collecting data from an SMF data set (Part 2 of 2)

Some logs require special collect procedures, which are supplied by the product. They are:

Collect job name

Description

DRLJCOIM

Collects IMS log data.

DRLJCOIN

Collects Tivoli Information Management for z/OS data.

DRLJCOVP

Collects network configuration data.

Collecting data from IMS

About this task

DRLJCOIM is a sample job for collecting data from the IMS SLDS log. For information about collecting IMS data and generating composite data records that combine various types of IMS log records, refer to the *IMS Performance Feature Guide and Reference*.

Collecting data from Tivoli Information Management for z/OS

About this task

The sample job, DRLJCOIN uses DRLJRFT2 to read data from the Tivoli Information Management for z/OS database. DRLJRFT2 is a Tivoli Information Management for z/OS report format table (RFT) in the DRLxxx.SDRLCNTL library. For information about collecting data from the Tivoli Information Management for z/OS database, refer to the *System Performance Feature Reference Volume 1*.

Collecting network configuration data

About this task

DRLJCOVP is a sample job for collecting network configuration data (vital product data). For information about collecting network configuration data, refer to the *Network Performance Feature Reference*.

Performing routine data collection

About this task

When you set up your collect jobs, consider these guidelines:

- Collect data at off-peak hours.
Log data sets are generally available, online systems have been taken down, and there is less contention for processing resources.
- Collect data daily, at least in the beginning (and especially from SMF and IMS logs).
- If you collect data from several systems, establish a procedure to get all the log data into the system that contains the product database.
- Set up automatic procedures for submitting collect jobs. For example, use Tivoli Workload Scheduler for z/OS (previously known as OPC, Operation Planning and Control) to initiate collect jobs. Refer to the Tivoli Workload Scheduler for z/OS documentation for more information about the product. You can also use the log data manager option to automate and obtain better control of the submitting of collect jobs. This option is described in Chapter 10, “Working with the log data manager option,” on page 205.

Monitoring collect activity

About this task

Tivoli Decision Support for z/OS provides statistics about collect activity in messages called *collect messages* and in the DRLSYS.DRLLOGDATASETS system table, described in the following sections.

Review collect activity to identify:

- Tables in high demand during collect processing. These tables are candidates for tuning to improve performance.
- Errors that occur in user-defined Tivoli Decision Support for z/OS objects.
- Any other errors that the log collector finds.

Sample collect messages:

Figure 28 shows a set of sample messages generated during a collect job.

```

DRL0300I Collect started at 2000-12-04-10.04.15
DRL0302I Processing SMF.DATA.SET on VOL001
DRL0341I The first record timestamp is 2000-06-03-07.00.01.730000.
DRL0308I A database update started after 2608 records due to a buffer-full condition
DRL0342I The last record timestamp is 2000-06-03-11.52.40.220000.
DRL0310I A database update started after 4582 records due to end of log
DRL0313I The collect buffer was filled 1 times. Consider increasing the
collect buffer size.
DRL0003I
DRL0315I Records read from the log or built by log procedure:
DRL0317I Record name          Number
DRL0318I -----
DRL0319I SMF_000                0
DRL0319I SMF_006                6
DRL0319I SMF_007                0
DRL0319I SMF_021                0
DRL0319I SMF_025                0
DRL0319I SMF_026                476
DRL0319I SMF_030                3737
DRL0319I SMF_070                40
DRL0319I SMF_071                40
DRL0319I SMF_072_1              280
DRL0319I SMF_090                0
DRL0320I Unrecognized          3
DRL0318I -----
DRL0321I Total                  4582
DRL0003I
DRL0316I Records built by record procedures:
DRL0317I Record name          Number
DRL0318I -----
DRL0319I SMF_030_X                 2012
DRL0319I SMF_070_X                 200
DRL0318I -----
DRL0321I Total                  2212
DRL0003I
DRL0323I
DRL0324I Table name          |-----Buffer-----|-----Database-----
DRL0325I |-----|-----|-----|-----|-----|-----
DRL0326I DRL .AVAILABILITY_D   |          3          23          2          1
DRL0326I DRL .AVAILABILITY_M   |          3           1          2          1
DRL0326I DRL .AVAILABILITY_T   |          9          76          9          0
DRL0326I :
DRL0326I DRL .MVS_WORKLOAD_H     |         144         336         132         12
DRL0326I DRL .MVS_WORKLOAD_M     |          60          12          48          12
DRL0325I |-----|-----|-----|-----|-----|-----
DRL0327I Total                |         2643        99019        2148         495
DRL0003I
DRL0301I Collect ended at 2000-12-04-10.09.43
DRL0356I To update the database, the algorithm SCAN was most selected.

```

Figure 28. Sample collect messages

Using collect messages:

About this task

To use collect messages effectively, follow this procedure:

Procedure

1. Identify which log was collected and when it started.

The first messages in a set of collect messages show when the collect starts and identify the data set. The product then shows the timestamp of the first identified record in the log, which looks like this:

```
DRL0341I The first record timestamp is
2000-06-03-07.00.01.730000.
```

2. Look for database activity.

The product writes data to the database when:

- The buffer is full. See “Improving collect performance” on page 94 if the buffer fills often. An example message is:

```
DRL0308I A database update started after 2608 records
due to a buffer-full condition
```

- All log data set records have been processed. An example message is:

```
DRL0310I A database update started after 4582 records
due to end of log
```

- A specific number of records have been read. The number is specified in the COMMIT AFTER operand of the COLLECT statement. An example message (where 1000 was specified as the COMMIT AFTER operand) is:

```
DRL0309I A database update started after 1000 records.
```

3. Determine the last record that the product identified in the log.

```
DRL0342I The last record timestamp is
2000-06-03-11.52.40.220000.
```

4. Review record-type statistical messages.

Collection statistics for record-type processing include:

- The type of each record processed
- The number of each record type found in the log data set
- The total number of records processed

Tivoli Decision Support for z/OS does not process any log records whose record type is either not defined, or defined but not used by collect. It issues a statistical message that labels the records *unrecognized*:

```
DRL0315I Records read from the log or built by log procedure:
```

Record name	Number
-----	-----
⋮	
DRL0319I SMF_026	476
DRL0319I SMF_030	3737
⋮	
DRL0320I Unrecognized	3
DRL0318I -----	-----
DRL0321I Total	4582

5. Verify that user-defined log, record, and update definitions are performing as expected. Check that appropriate data is being collected and stored in the appropriate tables.
6. Examine the processing performed by log and record procedures.

When Tivoli Decision Support for z/OS finds records that require handling by record procedures, it produces temporary, intermediate records for further Tivoli Decision Support for z/OS processing. Messages show the names and

Collecting log data

numbers of intermediate records built by record procedures while Tivoli Decision Support for z/OS was processing the log data set.

The messages appear in a group; for example:

```
DRL0316I Records built by record procedures:
DRL0317I Record name          Number
DRL0318I -----
DRL0319I SMF_030_X            2012
DRL0319I SMF_070_X            200
DRL0318I -----
DRL0321I Total                  2212
```

7. Examine database activity to identify tables with the most activity during collect processing.

Database inserts and updates show the number of rows inserted or updated in DB2 tables. The number of rows inserted in the database and the number of rows updated in the database equal the number of buffer inserts. Statistical messages of this sort look like these:

```
DRL0323I
DRL0324I Table name          |-----Buffer-----|-----Database-----
DRL0325I -----            | Inserts  Updates  | Inserts  Updates
DRL0326I DRL   .AVAILABILITY_D |          3      23  |          2      1
DRL0326I DRL   .MVS_WORKLOAD_M |         60      12  |         48      12
DRL0325I -----            |-----
DRL0327I Total                |        2643    99019 |        2148    495
```

8. You can use message DRL0356I to optimize the collect process by selecting the SCAN or DIRECT parameter. For more details, refer to the *Language Guide and Reference*, SH19-6817. Following is an example of message DRL0356I

```
DRL0356I To update the database, the algorithm SCAN was most selected.
```

Reviewing log statistics:

About this task

Use the administration dialog to create a log statistics file for any log data set, regardless of whether it has been collected. See “Displaying log statistics” on page 148 for more information.

Note: There are no lookup tables in the table name list.

Using the DRLLOGDATASETS table:

About this task

The DRLSYS.DRLLOGDATASETS system table contains one row of information for each log data set Tivoli Decision Support for z/OS collects. DRLLOGDATASETS contains collect statistics, such as elapsed time for a collection, record types collected, and numbers of records processed.

The product uses the data set name, log type, and the first 80 bytes from the first recognized record, to warn against attempts to collect a log data set already collected.

Data sets can contain identical records, but with different names. If you want to be notified when the second data set is processed, redefine the DRLLOGDATASETS system table so that it does not use the DATASET_NAME column as a key. Collection of the second data set fails with ABEND U0016 and an SQL code -803 against the DRLLOGDATASETS system table.

To view collect statistics, select a log definition from the Logs window, press F6 to see the data sets that have been collected for the log, choose a data set, and press Enter. The Collect Statistics window is displayed (Figure 29).

Note: *First timestamp* is the first record selected, *Last timestamp* is the last record selected. *Last timestamp* might show an earlier date and time than the first timestamp.

```

                                DCOLLECT Collect Statistics

Press Enter to return.

Data set . . . . . : IM3.DCOLLECT.SLOG14
Volume . . . . . : TSOL06

Time collected . . : 2000-02-11-12.38.00   Collected by . . . : LASZLOM
Elapsed time . . . : 54                   Return code . . . . : 4
Times collected . . : 3                   Completed . . . . . : Y

First record . . . : 000000700000E540000ID5D9C4F10048D2740092
                          276F0000000D7D9C9F0F0E70000000280010
First timestamp . . : 2000-10-02-13.15.24
Last timestamp . . : 2000-10-02-13.15.24

Records read . . . : 16458                Records selected . . : 16458

Database updates . : 7                   Inserts . . . . . : 4954       Deletes . . . . . : 0

F1=Help          F2=Split          F9=Swap          F12=Cancel

```

Figure 29. Collect Statistics window

Tivoli Decision Support for z/OS can produce a report from DRLLOGDATASETS that shows statistics for every collect job in the table.

The product does not update DRLLOGDATASETS until a collection results in a successful commit. If it finds an error that terminates processing of a log data set, such as a locking error or an out of space error, it does not update DRLLOGDATASETS. If it has already created a row for the log data set (which it does at the first commit), it does not update such indicators of a successful conclusion to processing as the Elapsed seconds column or the Complete column. See “Recovering from database errors” on page 107 for more information.

Refer to “DRLLOGDATASETS” on page 227 for a description of its columns.

Collecting multiple log data sets:

About this task

To collect multiple log data sets, specify the log data set names in the DRLLOG job card of the collect job as follows:

```

//DRLIN DD *
          COLLECT log-name
          ...
//DRLLOG DD DISP=SHR,DSN=log-dat-set-1
          DD DISP=SHR,DSN=log-data-set-2
          DD DISP=SHR,DSN=log-data-set-3
//DRLOUT DD SYSOUT=*

```

Collecting log data

If the log collecting job stops prematurely, you can start it again. In this case, the log collector does not collect the records of the data sets that were already completely processed and the following messages are issued:

```
DRL0302I Processing log-data-set-1 on EPDM0F
DRL0303W The log data set has already been processed. Data set name: log-data-set-1
```

The COLLECT process completes with a return code of 4.

If a log data set was only partially processed, the log collector does not collect the records that were already collected. In this way, the same data is not summarized twice.

Note: If the IMS checkpoint mechanism (DRLICHKI, DRLICHKO) is used, you cannot resubmit the same collect job when using multiple concatenated IMS data sets. If you resubmit the same collect job you could encounter a problem of duplicate key, because the DRLICHKI of the previous job would be used.

Improving collect performance

About this task

Correct collect performance problems using these tuning tasks:

Procedure

1. Optimize the collect buffer size. Optimizing the size of the collect buffer has the greatest impact on performance
 - a. Reduce the number of times Tivoli Decision Support for z/OS stops reading a log data set to write data to the database by increasing the buffer size.
Message DRL0313I shows the number of database updates because of a full buffer. Look for cases where the number of updates could be reduced by increasing the size of the buffer.
The optimum is to reduce the number of updates to 0.
 - b. The default buffer size is 10 MB. Use the buffer size operand of the COLLECT statement to increase the size to 20 MB to 30 MB, or more. Refer to the *Language Guide and Reference* for more information about the COLLECT statement.
 - c. Do not use the COMMIT AFTER nn records operand on the COLLECT statement.
2. Reduce the amount of data committed to the database.
 - a. Remove unnecessary tables using the INCLUDE/EXCLUDE clauses of the COLLECT statement.
 - b. Examine collect messages to determine the most active tables.
 - c. Concentrate on tables with a lot of buffer and database inserts and updates shown in DRL0326I messages.
 - d. Modify update definitions to eliminate needless rows in tables.
For example, set a key column to a constant (such as a blank) instead of to a value from a record if the detail is unnecessary.
 - e. Reduce the number of columns collected
 - 1) Delete unneeded columns from the update definition of the table.
 - 2) Remove the columns in the SQL CREATE TABLE statement of the table definition.
 - 3) Drop the table.

- 4) Re-create the table.

Note: Tivoli Decision Support for z/OS Version 1.8.2 makes use of the DB2 Version 8 multiple insert functionality. When data is collected to data tables, the insert statements are issued in bulk - 50 rows are inserted with a single DB2 multiple insert statement. This results in significant performance improvements. However, this performance improvement decreases as the number of columns inserted increases.

3. Improve update effectiveness.
 - a. Define an index on the primary key but no other indexes for tables you create.
 - b. Do not use a LOOKUP expression with the LIKE operand (especially for large lookup tables) in update definitions you create. Use an = operand where possible.
 - c. Minimize the number of rows in lookup tables that allow global search characters and in the PERIOD_PLAN control table.
 - d. Run collect when the processing load from other programs is low and when DB2 use is light.
 - e. Optionally, choose the appropriate algorithm to update the DB2 database by specifying the DIRECT or SCAN parameter in the COLLECT statement.

If you do not specify any parameter, the collect process automatically chooses an algorithm among the DIRECT, SCAN, and INSERT algorithms. This automatic selection, however, can be very time consuming. To improve the performance, you can force the collect process to use either the DIRECT or SCAN algorithm only, by specifying the DIRECT or SCAN parameter in the COLLECT statement. For details about these parameters, refer to the *Language Guide and Reference*, SH19-6817 manual.

Administering the Tivoli Decision Support for z/OS database

About this task

Maintaining the Tivoli Decision Support for z/OS database includes purging unneeded data, reorganizing the database, updating DB2 statistics, backing up data, updating views on the DB2 catalog, and protecting the integrity of data by controlling the access.

Regular maintenance tasks are as follows:

Besides regularly scheduled jobs, run the RUNSTATS utility periodically while the database is growing to:

- Provide the DB2 optimizer with information. (After the database stabilizes, RUNSTATS does not make a significant contribution to the DB2 optimizer.)
- Provide table size statistics for Tivoli Decision Support for z/OS.

See “Monitoring the size of the Tivoli Decision Support for z/OS database” on page 108 for more information.

The remainder of this section introduces the use of DB2 as the product database manager and shows how to use DB2 to maintain the database.

Understanding DB2 concepts

By default, Tivoli Decision Support for z/OS names for DB2-related items are:

Administering the Tivoli Decision Support for z/OS database

Tivoli Decision Support for z/OS name

Description

DSN Names the DB2 subsystem.

DRLDB

Names the product database.

DRLSSYS1

Names the product table space that contains log collector system tables.

DRLSSYS2

Names the product table space that contains other system tables.

DRLSSAMP

Names the product table space that contains tables for the Sample component

DRLSCOM

Names the product table space that contains common tables that the product components use.

The names of other product table spaces depend on the components you install. There is at least one table space for each component.

Figure 30 shows the data areas in the DB2 subsystem.

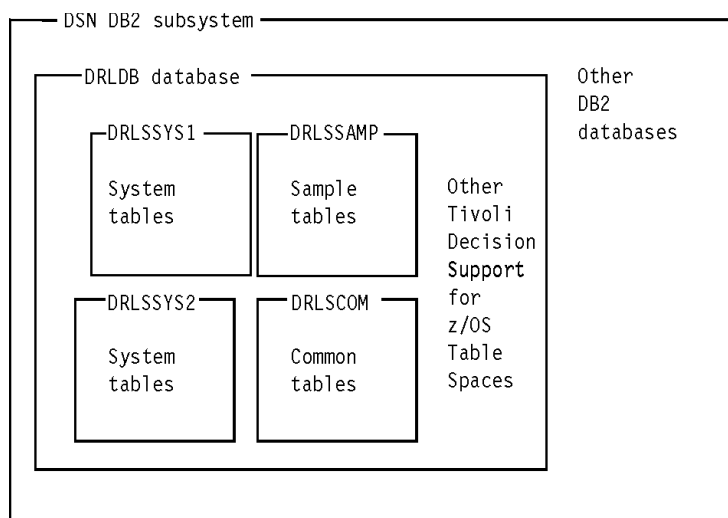


Figure 30. DB2 environment for the Tivoli Decision Support for z/OS database

Understanding how Tivoli Decision Support for z/OS uses DB2

Figure 30 shows a Tivoli Decision Support for z/OS installation that uses one product database. There can be more than one product database in the installation of Tivoli Decision Support for z/OS, more than one Tivoli Decision Support for z/OS installation in one DB2 subsystem, more than one DB2 subsystem with an installation of the product, and so on.

Understanding table spaces

Figure 30 shows that the product uses several table spaces in the DRLDB database. A table space contains one or more tables and is the logical unit addressed by DB2 utilities such as COPY and REORGanize.

The DRLSSYS1 and DRLSSYS2 table spaces contain Tivoli Decision Support for z/OS system tables and always exist in a functioning product system. When you install a Tivoli Decision Support for z/OS component, it creates at least one

Administering the Tivoli Decision Support for z/OS database

segmented table space for the component within its database. The exact configuration of table spaces you have depends on the components you have installed.

To list the table spaces belonging to the current database:

1. Select 4, Tables, from the Administration window.
2. Without selecting a table, select the Maintenance pull-down.
3. Select 1, Tablespace, from the options.

Figure 31 shows the list of table spaces, with the Utilities pull-down.

The screenshot shows a window titled 'Tablespace list window'. At the top, there are four menu options: 'Tablespace', 'Utilities', 'Other', and 'Help'. The 'Utilities' menu is open, showing three options: '1. Run DB2 REORG utility...', '2. Run DB2 RUNSTATS utility..', and '3. Run DB2 REORG/DISCAD utility..'. To the right of the menu, it says 'Row 1 to 20 of 37' and 'ablespace definition.'. Below the menu, there is a table with the following columns: 'Tablespace', 'Primary', 'Secondary', 'Storage grp', 'Type', and 'Locksize'. The table lists several table spaces with their respective primary and secondary quantities and storage groups.

Tablespace	Primary	Secondary	Storage grp	Type	Locksize
DRLSAIX	6000	3000	SYSDEFLT	SEGMENTED	TABLE
DRLSCI08	100	52	STOEPDM	SEGMENTED	TABLE
DRLSCOM	20000	10000	SYSDEFLT	SEGMENTED	TABLE
DRLSCP	60	32	SYSDEFLT	SEGMENTED	TABLE
DRLSDB2	40000	20000	SYSDEFLT	SEGMENTED	TABLE
DRLSDFSM	60000	30000	SYSDEFLT	SEGMENTED	TABLE
DRLSDPAM	100	52	SYSDEFLT	SIMPLE	ANY

Figure 31. Tablespace list window

When you change table space or indexspace parameters, the product uses SQL commands to alter the space directly, and creates a job to unload and load table data as necessary. Tivoli Decision Support for z/OS does not change the **definition** of the table space. To do this, select the Space pull-down on the Components window.

If you create a table in the product database, you must specify the database and table space in which DB2 is to create the table. Once created, a table can be addressed by its table name only. You do not need to specify the table space name.

“Working with tables and update definitions” on page 179 describes how to use the administration dialog to view, change, or create table spaces.

Calculating and monitoring table space requirements

About this task

To make effective use of the available space, you need to monitor the storage required for your data tables. The sample job, DRLJTBSR (in the DRL182.SDRLCNTL library), produces a detailed report about the space required for some or all of the selected component tables, based on the average record size and estimated number of rows.

To customize the job to your requirements, you must change some parameters in DRLJTBSR. For a description of these parameters, see “Parameters for table space reporting” on page 99.

Administering the Tivoli Decision Support for z/OS database

DRLJTBSR job that reports tablespace requirements

```
//DRLJTBSR JOB (ACCT#),'SPACE'                                00000100
//*****                                                    00000200
//*                                                         * 00000300
//* Licensed Materials - Property of IBM                    * 00000400
//*                                                         * 00000500
//* 5698-B06 Copyright IBM Corporation 1992, 2015         * 00000600
//* See Copyright instructions.                             * 00000700
//*                                                         * 00000800
//*****                                                    00000900
//*                                                         * 00001000
//* Name: DRLJTBSR                                          * 00001100
//*                                                         * 00001200
//* STATUS: Tivoli Decision Support for zOS 1.8.2          * 00001300
//*                                                         * 00001400
//* FUNCTION: Print a report of estimated total kilobytes based on
//*             estimated records number and average record length * 00001500
//*             for each table on component.                * 00001600
//*             Average records length is calculated,if the table is
//*             not created, reading TDS for zOS           * 00001700
//*             definition                                  * 00001800
//*             library                                    * 00001900
//*                                                         * 00002000
//* The exec DRLETBSR accepts the following parameters:    * 00002100
//*                                                         * 00002200
//* LIBRARY= TDS for zOS definition library                 * 00002300
//* SYSPREFIX= TDS for zOS system table prefix             * 00002400
//* DB2SUBSYS= Db2 subsystem name                          * 00002500
//* COMPONENT= Component name. To have a complete list of * 00002600
//*             component short name read the DRLCOMPONENTS * 00002700
//*             system table                                * 00002800
//* TABLENAME= Table name ('*' to select all table)      * 00002900
//* RECNUMBER= Estimated record numbers                    * 00003000
//* PAGESIZE= Value of pagesize . Can be 4K or 32K.        * 00003100
//*             Optional parameter, default value when     * 00003200
//*             not specified is 4K                         * 00003300
//* MAXROWS= Maximum number of rows per pages. Maximum    * 00003400
//*             value allowed is 255.                      * 00003500
//*             Optional parameter, default value when not * 00003600
//*             specified is 255                            * 00003700
//* PCTFREE= Percentage of free space on each page.        * 00003800
//*             Value allowed from 0 to 99.                 * 00003900
//*             Optional parameter, default value when not * 00004000
//*             specified is 5                              * 00004100
//* FREEPAGE= Number of free space pages. Value allowed    * 00004200
//*             from 0 to 255.                              * 00004300
//*             Optional parameter, default value when not * 00004400
//*             specified is 0                              * 00004500
//* COMPRESS= Compression ratio. Optional parameter.       * 00004600
//*             The value must be in range from 0 to a value * 00004700
//*             less than 1                                 * 00004800
//*             Default value when not specified is 0.      * 00004900
//* Notes:                                                  * 00005000
//* Before you submit the job, do the following:           * 00005100
//* 1. Check that the data set names are correct.          * 00005200
//* 2. Change the parameters to DRLETBSR as required.      * 00005300
//* 3. Change the DB2 load library name according to       * 00005400
//*    the naming convention of your installation.         * 00005500
//*    Default is 'db2loadlibrary'.                        * 00005600
//* CHANGE ACTIVITY:                                       * 00005700
//* 00 2015-12-22 SL Created PTR153                        * 00005800
//*                                                         * 00005900
//*                                                         * 00006000
//* CHANGE ACTIVITY:                                       * 00006100
//* CHANGE FLAG TYPE DATE DESCRIPTION                     * 00006200
//* -----*
//* $D0=DCR066, TDS182,01/06/15,ADL(SM): Update TDS Version and * 00006300
//*             DB2 dataset names.                          * 00006400
//* $D1=DCR116, TDS182,15/05/15,ADL(RC): Update TDS Version * 00006500
//*                                                         * 00006600
//*                                                         * 00006700
//*****                                                    00006800
//SPACE EXEC PGM=IKJEFT01,DYNAMNBR=25                     00006900
//*                                                         00007000
//STEPLIB DD DISP=SHR,DSN=DRLvrms.SDRLLLOAD <== DATA SET NAME 00007100
//SYSPROC DD DISP=SHR,DSN=DRLvrms.SDRLEXEC <== DATA SET NAME 00007200
//SYSEXEC DD DISP=SHR,DSN=DRLvrms.SDRLEXEC <== DATA SET NAME 00007300
//*****                                                    00007400
//* START EXEC DRLETBSR                                     00007500
```

Administering the Tivoli Decision Support for z/OS database

```

//SYSPRINT DD SYSOUT=*                                00007600
//SYSTSPRT DD SYSOUT=*                                00007700
//SYSYSIN  DD *                                        00007800
%DRLETBSR LIBRARY= DRLvrm.SDRLDEFS                    -      00007900
          DB2SUBSYS= DSN                               -      00008000
          SYSPREFIX= DRLSYS                           -      00008100
          COMPONENT= xxxx                             -      00008200
          TABLENAME= *                               -      00008300
          RECNUMBER= xxxx                             -      00008400
          PAGESIZE= 4K                                -      00008500
          MAXROWS= 255                                -      00008600
          PCTFREE= 5                                  -      00008700
          FREEPAGE= 0                                 -      00008800
          COMPRESS= 0                                  -      00008900
/*                                                    00009000
    
```

Following is sample output for job DRLJTBSR that shows the space required for all tables of the IMS collect component.

Sample output for DRLJTBSR

Statistics for space required for a component:
-----;

```

Input library      : DRL182.SDRLDEFS
Db2 subsystem     : DSN7
PR system prefix  : PRM3SYS
Component         : IMSV710C
Table name        : *
Estimated records number : 500000
Page size         : 4096
Maxrows per page  : 255
Percentage of free space : 5
Number of free pages : 0
Compression ratio  : 0
    
```

Table name	New	Tablespace	Definition member	Avg record length	Record per page	Estimated total pages	Estimated kilobytes
IMS_APPLICATION_H	N	DRLSIA01	DRLTIMSA	651	5	100002	400008
IMS_APPLICATION_W	N	DRLSIA02	DRLTIMSA	648	5	100002	400008
IMS_CHKPT_IOSAM_T	N	DRLSIS01	DRLTIMSS	169	22	22730	90920
IMS_CHKPT_POOLS_T	N	DRLSIS02	DRLTIMSS	99	39	12823	51292
IMS_CHKPT_REGION_T	N	DRLSIS03	DRLTIMSS	101	38	13160	52640
IMS_CHKPT_STATS_T	N	DRLSIS04	DRLTIMSS	518	7	71430	285720
IMS_CHKPT_VSAM_T	N	DRLSIS05	DRLTIMSS	194	19	26318	105272
IMS_SYSTEM_D	N	DRLSIY01	DRLTIMSY	642	6	83335	333340
IMS_SYSTEM_Q	N	DRLSIY02	DRLTIMSY	645	6	83335	333340
IMS_TRANSACTION_D	N	DRLSIT02	DRLTIMSR	646	5	100002	400008
IMS_TRANSACTION_H	N	DRLSIT01	DRLTIMSR	649	5	100002	400008
IMS_TRANSACTION_W	N	DRLSIT03	DRLTIMSR	646	5	100002	400008

Parameters for table space reporting

Table 6. Parameters for tablespace reporting

Parameter	Value to set	Explanation	Default value	Your value
LIBRARY	Tivoli Decision Support for z/OS definition library (UPPERCASE)	The name of the partitioned data set that contains definitions of the product tables. This is a required parameter. It is used for component tables that do not yet exist.		
DB2SUBSYS	DB2 subsystems name (UPPERCASE)	The DB2 subsystem where the product resides. This is a required parameter.		
SYSPREFIX	Prefix for system tables (UPPERCASE)	The prefix of all Tivoli Decision Support for z/OS system and control DB2 tables. This is a required parameter. The value of this parameter depends on your naming convention and is determined during installation.		

Administering the Tivoli Decision Support for z/OS database

Table 6. Parameters for tablespace reporting (continued)

Parameter	Value to set	Explanation	Default value	Your value
COMPONENT	Component name (UPPERCASE)	The name of a Tivoli Decision Support for z/OS component. This is a required parameter.		
TABlename	The name of the table (UPPERCASE)	The name of the Tivoli Decision Support for z/OS table. This is a required parameter. To specify all component tables, type an asterisk, *. To specify all component tables whose names start with a particular string, type the string. For example, type CICS_S for all component tables whose name starts with this string.		
RECNUMBER	Number of rows	The estimated number of rows. This is a required parameter and must be numeric.		
PAGESIZE	DB2 page size	The DB2 page size. This is an optional parameter; when specified, it must be either 4K or 32K.	4096 (4K)	
MAXROWS	Maximum number of rows per page	The maximum number of rows per page. This is an optional parameter; when specified, it must be a numeric value between 1 and 255.	255	
PCTFREE	Percentage of free space on each page	The percentage of free space per page. This is an optional DB2 parameter; when specified, it must be a numeric value between 1 and 255.	5	
FREEPAGE	Number of free space pages	The number of free space pages. This is an optional DB2 parameter; when specified, it must be a numeric value between 1 and 255.	0	
COMPRESS	Compression ratio	The compression ratio calculated as PERCSAVE/100 (PERCSAVE is the percentage of kilobytes saved by compression as reported by DB2 utility DSN1COMP). This parameter is optional; when specified, it must be a numeric value.	0	

For detailed information about the parameters, refer to the *DB2 Universal Database for OS/390 and z/OS: SQL Reference*.

For information about DB2, refer to the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

For information about the algorithm used for calculating table space requirements, refer to the *DB2 for OS/390 Installation Guide*.

Considerations when running DRLJTBSR

The sample job DRLSJTBSR invokes the DRLETBSR exec. Before you can use DRLETBSR, the Tivoli Decision Support for z/OS system tables must have already been created or updated. If a component is already installed, DRLETBSR obtains the average record size of each component table directly from the product system tables.

The column NEW in the report shows the table status (N for a table already created, Y for a table that does not exist). The DRLETBSR exec calculates the average record size for each component table.

If a component is not installed, the DRLETBSR exec reads each partitioned data set member that defines each component table (see the LIBRARY parameter). Use this

Administering the Tivoli Decision Support for z/OS database

exec only for standard Tivoli Decision Support for z/OS libraries. Using it for customized libraries can produce unpredictable results. For variable length fields, the average record size is calculated using the maximum length. The average record size does not include GRAPHIC, VARGRAPHIC and LONG VARGRAPHIC DB2 data-types. When you specify the estimated number of records, remember that the product collects data from tables according to rules specified in the update definitions. Tables containing the same data may therefore have different numbers of rows. For example, an hourly table may contain a greater number of rows than a daily table.

Reorganizing the database

It is important to delete obsolete data from the tables as this updates the product database and improves performance during the query activity. Also, it is important to reorganize table space after data deletion to optimize the available space. You can use the utility Reorg/Discard to delete data and reorganize table space.

Reorg/Discard utility

The Reorg/Discard utility enables you to delete the data included in the tables using the Purge condition included in the DRLPURGECOND table. This table is provided in Tivoli Decision Support for z/OS. At the same time, the Reorg/Discard utility automatically reorganizes the table space where data has been deleted.

The records deleted by the Discard function are automatically saved in a specific data set, SYSPUNCH. SYSPUNCH can be used at a later time to reload discarded data in the table, if required.

During the Discard step, the Reorg function reorganizes the table space to improve access performance and reclaim fragmented space. Also, the keyword STATISTICS is automatically selected for the Reorg/Discard, enabling you to collect online statistics during database reorganization.

See the *DB2 Universal Database for OS/390 and z/OS: Utility Guide and Reference*, for more information about Reorg/Discard utility.

There are two ways to run the Reorg/Discard utility from the Administration window of Tivoli Decision Support for z/OS:

From the Tables window, select option 12 from the Utilities pull-down menu.

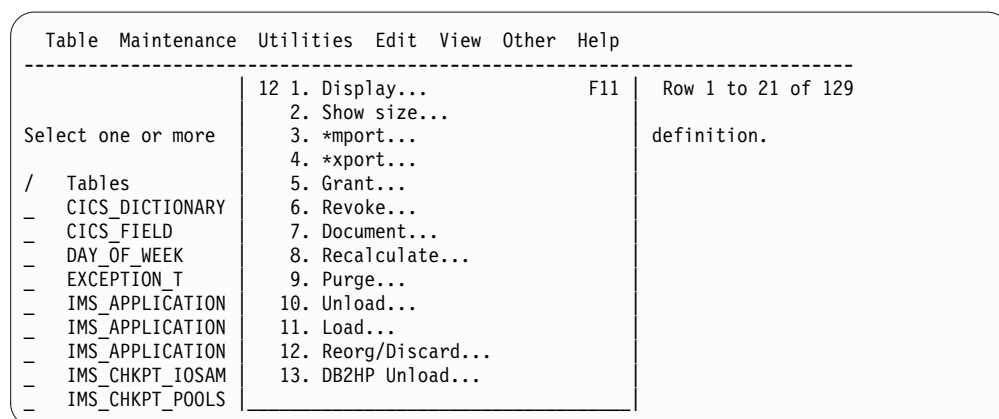


Figure 32. Tables window - Option 12

Administering the Tivoli Decision Support for z/OS database

In this way, the data contained in the table or tables selected from the table list is discarded, and a space reorganization is automatically performed in the table space where the selected tables reside. The Discard operation is only performed on the selected tables, while the Reorg operation is performed on all the tables contained in the table space. You cannot run the Discard utility on Views or Tables that have any discard condition specified in the DRLPURGECOND table.

As an alternative, use option 1 from the Maintenance pull-down menu of the Tables window to open the Tablespace window, then select option 3 from the Utilities pull-down menu.

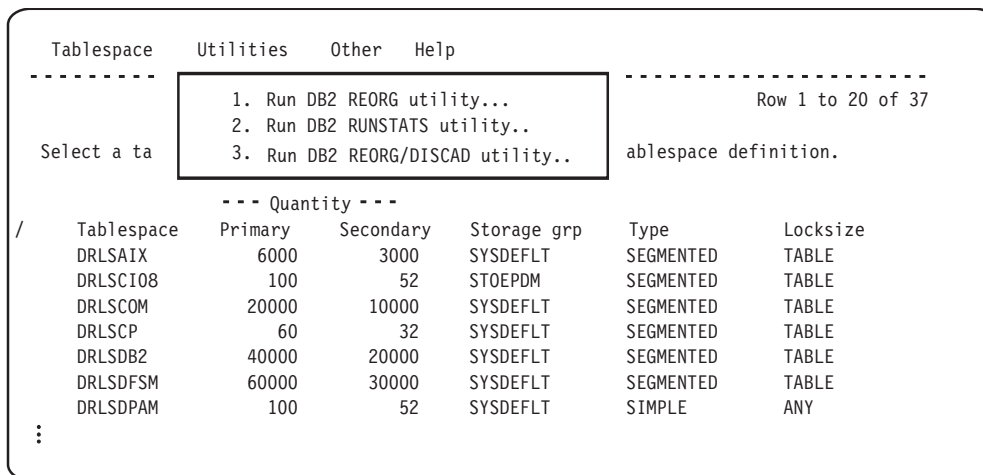


Figure 33. Tablespace list window

In this second scenario, from the Tablespace window, you select the table spaces for the Reorg operation. The Discard operation is automatically run on all the tables contained in the selected table spaces, according to the conditions specified in the DRLPURGECOND table.

All the tables that have the Discard operation specified in the DRLPURGECOND table are included in the processing. All the tables that do not have the Discard operation specified in the DRLPURGECOND table are ignored.

When you run Reorg/Discard, whichever procedure you use, a JCL is created and saved in your library, so that it can be used at a later time, if required. When the JCL is launched, the following data sets are created:

SYSPUNCH

Used to reload the discarded data, if required, using the Load utility.

SYSDISC

Contains the records discarded by the utility.

In addition, SYSREC data set is available. It contains all the records in the table, and you can specify whether you want it to be Temporary or Permanent. If you specify Temporary, the data set is automatically erased at the end of the reorganization job. If you specify Permanent, it is permanently allocated on your disk.

Administering the Tivoli Decision Support for z/OS database

When using the Reorg/Discard utility, you can select one or more tables and table spaces at a time. However, in the data sets SYSPUNCH and SYSDISC, data is overwritten, therefore each data set maintains only the information contained in the last table you processed.

The following is an example of how the Reorg/Discard utility works on a table space that contains several tables:

```
//REODIS JOB (ACCOUNT),'NAME'
//*
//*****
//* Run DB2 Utility
//*
//* WARNING (REORG/DISCARD):
//* If you want, you can specify the SORTKEYES option:
//* a subtask sorts the index keys. For this optional
//* operation you have need of enough space in your
//* default Storage Diskfor this SORT operation.
//*
//*****
//DB2UTIL EXEC DSNUPROC,
// SYSTEM=DSN6,UID=MYUID
//*
//DSNUPROC.STEPLIB DD DISP=SHR,DSN='db2loadlibrary'
//DSNUPROC.SYSREC DD DSN=MYUID.DRLUNLD,UNIT=SYSDA,
// SPACE=(4096,(1,1)),DISP=(MOD,DELETE,CATLG)
//DSNUPROC.SYSUT1 DD DSN=MYUID.DRLWORK,UNIT=SYSDA,
// SPACE=(4096,(1,1)),DISP=(MOD,DELETE,CATLG)
//DSNUPROC.SORTOUT DD DSN=MYUID.DRLSROUT,UNIT=SYSDA,
// SPACE=(4096,(1,1)),DISP=(MOD,DELETE,CATLG)
//DSNUPROC.WORK DD DSN=MYUID.WORK1,UNIT=SYSDA,
// SPACE=(4096,(1,1)),DISP=(MOD,DELETE,CATLG)
//DSNUPROC.SYSPUNCH DD DISP=(MOD,CATLG),
// DSN=MYUID.TAB.SYSPUNCH,
// SPACE=(4096,(1,1)),UNIT=SYSDA
//DSNUPROC.SYSDISC DD DISP=(MOD,CATLG),
// DSN=MYUID.TAB.DISCARDS,
// SPACE=(4096,(5040,504)),UNIT=SYSDA,
// DCB=(RECFM=FB,LRECL=410,BLKSIZE=27880)
//DSNUPROC.SYSIN DD *
REORG TABLESPACE MYDB.DRLSCOM LOG YES
STATISTICS INDEX(ALL) DISCARD
FROM TABLE MYDB.AVAILABILITY_D
WHEN (
DATE < CURRENT DATE - 90 DAYS
)
FROM TABLE MYDB.AVAILABILITY_T
WHEN (
DATE < CURRENT DATE - 14 DAYS
)
FROM TABLE MYDB.AVAILABILITY_M
WHEN (
DATE < CURRENT DATE - 104 DAYS
)
/*
```

In this example, the Reorg/Discard utility reorganizes the MYUID.DRLSCOM table space and discards data from the MYDB.AVAILABILITY_D, MYDB.AVAILABILITY_M, and MYDB.AVAILABILITY_T tables. This example shows that the DDNAME for the SYSPUNCH data set is SYSPUNCH, the DDNAME for the discard results data set is SYSDISC, and the DDNAME for the sort output data set is defaulted to SORTOUT. The SYSDISC and SYSPUNCH data sets are reused every time the utility is run for all tables.

Administering the Tivoli Decision Support for z/OS database

Purge utility

As an alternative to the Reorg/Discard utility, you can delete data and reorganize table space using the Purge utility. Each data table in a component has a Purge condition that specifies which data is to be purged from that table. When you use the Purge function, the data specified in the purge condition is deleted.

Purge the contents of your database at least weekly. The sample job, DRLJPURG (in the DRL182.SDRLCNTL library), purges all product database tables with Purge conditions. Figure 34 shows part of DRLJPURG.

```
//DRLJPURG JOB (ACCT#),'PURGE'                                00010003
//*****                                                    00020000
//*                                                           *      00030000
//* LICENSED MATERIALS - PROPERTY OF IBM                    *      00040004
//*                                                           *      00050000
//* 5698-B06 Copyright IBM Corporation 1992, 2015          *      00060000
//* SEE COPYRIGHT INSTRUCTIONS.                             *      00070004
//*                                                           *      00080000
//*****                                                    00090000
//*                                                           *      00100000
//* NAME: DRLJPURG                                          *      00110004
//*                                                           *      00120000
//* STATUS: Tivoli Decision Support for zOS 1.8.2          *      00130004
//*                                                           *      00140000
//* FUNCTION:                                              *      00150004
//* PURGE DATA FROM ALL TDS for zOS TABLES               *      00160004
//* ACCORDING TO THE PURGE CONDITIONS DEFINED FOR THE TABLES)* 00170004
//* IF YOU WANT TO PURGE ONLY SOME TABLES, SPECIFY THE   *      00180004
//* INCLUDE OR EXCLUDE OPTIONS. EXAMPLE:                   *      00190004
//*                                                           *      00200000
//* PURGE INCLUDE LIKE 'DRL.CICS%'                          *      00210000
//*                                                           *      00220000
//* NOTES:                                                  *      00230004
//* 1.CHECK DB2 SUBSYSTEM AND DATA SET NAMES.             *      00240004
//* 2.Change the DB2 load library name according to        *      00270000
//* the naming convention of your installation.             *      00260000
//* Default is 'db2loadlibrary'.                           *      00260000
//*                                                           *      00250000
//*                                                           *
//* CHANGE ACTIVITY:                                       *
//* CHANGE FLAG TYPE DATE DESCRIPTION                      *
//* -----*
//* $D0=DCR066, TDS182,01/06/07,ADL(SM): Update TDS Version and *
//* DB2 dataset names.                                     *
//* $D1=DCR116, TDS182,15/05/09,ADL(RC): Update TDS Version *
//*                                                           *
//*****                                                    00260000
//PURGE EXEC PGM=DRLPLC,PARM=('SYSTEM=DSN SYSPREFIX=DRLSYS') 00270000
//STEPLIB DD DISP=SHR,DSN=DRLvrn.SDRLLoad                   00280002
// DD DISP=SHR,DSN=db2loadlibrary                           00290000
//DRLIN DD *                                                 00300000
//                                                           00310000
PURGE;                                                       00320000
//                                                           00330000
//DRLOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)                 00340000
//DRLDUMP DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)                00350000
//*                                                         00360000
```

Figure 34. DRLJPURG job that uses all purge conditions

The Purge utility generates messages that show if the job ran as expected:

```
DRL0300I Purge started at 2000-05-24-15.12.30.
DRL0404I Table name | Deletes
DRL0405I -----|-----
```


Administering the Tivoli Decision Support for z/OS database

```
DRL0406I DRL      .RACF_RESOURCE_T      |      12376
DRL0406I DRL      .RACF_LOGON_T         |           98
DRL0406I DRL      .RACF_OPERATION_T     |         457
DRL0406I DRL      .RACF_COMMAND_T       |          17
DRL0301I Purge ended at 2000-05-24-15.12.44.
```

After purging the database, use the DB2 REORG utility to free the purged space for future use. There are three methods of reorganizing your database:

1. Use option 1, Run DB2 REORG utility, from the Utilities menu on the Tablespace list window, shown in Figure 31 on page 97. This reorganizes a whole table space.
2. Use option 10, Unload, from the Utilities menu on the Tables window, after having selected one or more tables. When you Unload and then Load a table, it reorganizes it without affecting the other tables in the table space.

Figure 35 shows the list of tables, with the Utilities pull-down.

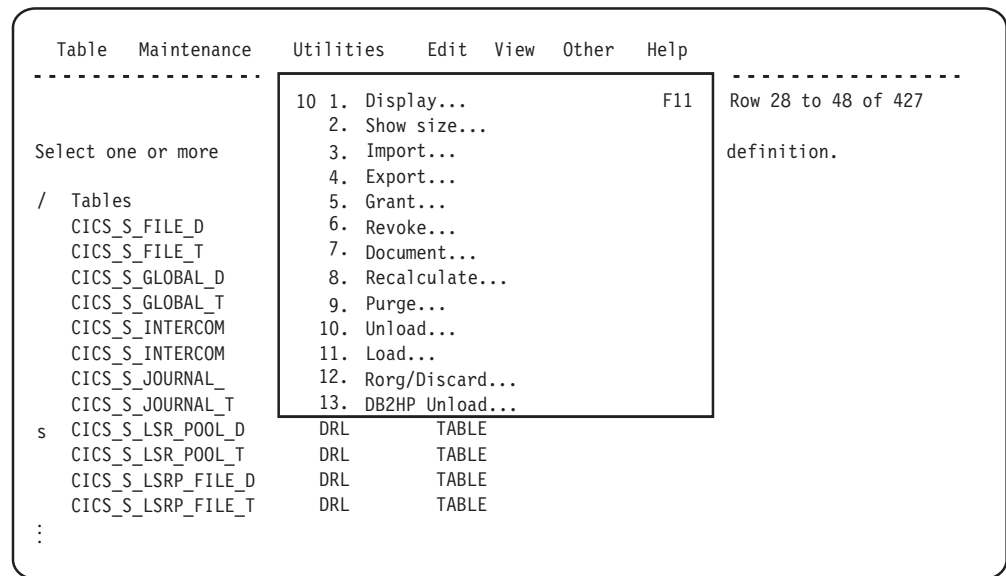


Figure 35. Tables window -Option 10

3. Use the sample job DRLJREOR (in the DRL182.SDRLCNTL library) to build your own job.

Refer to the description of the REORG utility in the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference* for more information.

Backing up the Tivoli Decision Support for z/OS database

About this task

Back up the Tivoli Decision Support for z/OS database regularly. Ask your DB2 administrator to add your requirements to their site DB2 procedures for backing up the data. If you cannot do this, copy and modify the sample job DRLJCOPY (in the DRL182.SDRLCNTL library), to back up all product tables.

Determine:

- How often to back up the product database
- Whether to back up all data or just changed data
- The names of table spaces in the database

Administering the Tivoli Decision Support for z/OS database

Figure 36 shows job DRLJCOPY, used to back up the DRLSSYS1 and DRLSSYS2 table spaces.

```

//DRLJCOPY JOB (ACCT#),'IMAGE COPY'                                00010001
//*****                                                             00020000
//*                                                                    * 00030000
//* LICENSED MATERIALS - PROPERTY OF IBM                            * 00040002
//*                                                                    * 00050000
//* 5698-B06 Copyright IBM Corporation 1992, 2015                  * 00060000
//* SEE COPYRIGHT INSTRUCTIONS.                                     * 00070002
//*                                                                    * 00080000
//*****                                                             00090000
//*                                                                    * 00100000
//* NAME: DRLJCOPY                                                 * 00110002
//*                                                                    * 00120000
//* STATUS: Tivoli Decision Support for zOS 1.8.2                  * 00130002
//*                                                                    * 00140000
//* FUNCTION:                                                       * 00150002
//*   RUN THE DB2 IMAGE COPY UTILITY TO MAKE BACKUP COPIES        * 00160002
//*   OF TDSzOS           TABLE SPACES. THIS JOB COPIES         * 00170002
//*   TABLE SPACES DRLSSYS1 AND DRLSSYS2. YOU MUST ADD A COPY * 00180002
//*   STATEMENT AND DATA SET FOR EACH TABLE SPACE THAT YOU    * 00190002
//*   WANT TO BACK UP.                                           * 00200002
//*                                                                    * 00210000
//* NOTES:                                                           * 00220002
//*   CHECK THE FOLLOWING:                                          * 00230002
//*     LIB='db2loadlibrary'      DB2 LOAD LIBRARY                * 00240002
//*     SYSTEM=DSN                 DB2 SUBSYSTEM NAME             * 00250002
//*     DSN=COPYDSN                NAME OF BACKUP DATASET         * 00260002
//*     SPACE=                     SPACE REQUIRED                  * 00270002
//*     COPY TABLESPACE DB.TS     DATABASE.TABLESPACE NAME     * 00280002
//*     FULL YES/NO                FULL OR INCREMENTAL COPY      * 00290002
//*                                                                    * 00300000
//*                                                                    *
//* CHANGE ACTIVITY:                                               *
//* CHANGE FLAG  TYPE    DATE    DESCRIPTION                       *
//* -----
//* $D0=DCR066, TDS182,01/06/15,ADL(SM): Update TDS Version and *
//*                                                                    *
//*                                                                    *
//*****                                                             00310000
//*                                                                    * 00320000
//*                                                                    * 00330000
//* UTIL EXEC DSNUPROC,LIB='db2loadlibrary',                       * 00340000
//*   SYSTEM=DSN,UID='TEMP',UTPROC=' '                             * 00350000
//*                                                                    * 00360002
//* COPY01 DD DSN=COPYDSN1,                                         * 00370000
//*   DISP=(MOD,CATLG,CATLG),                                       * 00380000
//*   SPACE=(16384,(50,50),,,ROUND),                                 * 00390000
//*   UNIT=SYSDA                                                       * 00400002
//* COPY02 DD DSN=COPYDSN2,                                         * 00410000
//*   DISP=(MOD,CATLG,CATLG),                                       * 00420000
//*   SPACE=(16384,(50,50),,,ROUND),                                 * 00430000
//*   UNIT=SYSDA                                                       * 00440000
//* SYSIN DD *                                                         * 00450000
COPY TABLESPACE DRLDB.DRLSSYS1                                     00460000
  COPYDDN COPY01                                                  00470000
  FULL YES                                                         00480000
COPY TABLESPACE DRLDB.DRLSSYS2                                     00490000
  COPYDDN COPY02                                                  00500000
  FULL YES                                                         00510000
//*

```

Figure 36. DRLJCOPY job for backing up Tivoli Decision Support for z/OS table spaces

Determining when to back up the Tivoli Decision Support for z/OS database

About this task

Back up the database at least weekly to make it easier to recover from errors.

Determining a level of backup

About this task

DB2 provides two methods for backing up data, full-image copy (copy all data), and incremental-image copy (copy only changed data). You can combine the two copies.

Determining which table spaces to back up

About this task

The DB2 COPY utility operates on tablespaces. Ensure that all tablespaces are part of the backup procedures. For more information about backing up a DB2 database, refer to the discussion of backing up and recovering databases in the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

Recovering from database errors

About this task

These errors might occur in a Tivoli Decision Support for z/OS database that sees significant activity:

- Out of space in one of the table spaces or index spaces
- Corrupted data in the database

The following sections contain descriptions of each condition, how it might occur, and how to correct it.

A description of how to restore DB2 database backups appears in “Correcting corrupted data in the Tivoli Decision Support for z/OS database” on page 108.

Correcting out-of-space condition in a Tivoli Decision Support for z/OS table space or index space

About this task

A table space or index space can be out of space if:

- Volumes in the Tivoli Decision Support for z/OS storage group are full.
If DASD is not constrained, the database continues to grow and performance can be an issue. If performance is not an issue, ask the DB2 administrator to add volumes to the Tivoli Decision Support for z/OS storage group.
If you cannot add more volumes to your storage group, purge the database before continuing. After purging data, reorganize the affected table spaces. See “Purge utility” on page 104 for more information.
- The table space or index space uses its maximum number of extents.

This scenario can occur if the primary quantity and all secondary quantity (PRIQTY and SECQTY) extents have been exhausted. Tivoli Decision Support for z/OS table spaces and index spaces have a default size specification based on an estimated number of rows in tables in the table space. These default values may be too small for a very large site.

Administering the Tivoli Decision Support for z/OS database

To recover from an out-of-space condition:

Procedure

1. Increase the primary and secondary quantities using the Tivoli Decision Support for z/OS administration dialog (Figure 90 on page 196), or by using the DB2 SQL statements, ALTER TABLESPACE or ALTER INDEX.
2. Reorganize the table space using the DB2 REORG utility as described in “Purge utility” on page 104 or drop the index and recreate it as described in “Displaying and adding a table index” on page 182.

```
DSNT408I  SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION
        CAUSED BY AN UNAVAILABLE RESOURCE. REASON
        00D70025, TYPE OF RESOURCE 00000220 AND RESOURCE
        NAME DB2A.DSNDBC.DRLDB.A.I0001.A001
```

Correcting corrupted data in the Tivoli Decision Support for z/OS database

About this task

Corrupted data can occur because of:

- DB2 errors
- Erroneously collecting the same log data set more than once

If the database has been incorrectly updated (for example, accidentally collecting the same log data set twice or deleting required data), restore a previous backup copy with the DB2 RECOVER utility. For information about backing up and recovering DB2 databases, refer to the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

You need not restore product data after a collect job terminates from locking or out of space. After correcting the error, run the job again. If the database has been updated, the collect resumes from the last checkpoint recorded in the DRLSYS.DRLLOGDATASETS system table. If it had not committed data to the database before the error, Tivoli Decision Support for z/OS recovers the data by collecting from the first record in the log.

Monitoring the size of the Tivoli Decision Support for z/OS database

About this task

Monitor the size of the database regularly. Use the DB2 RUNSTATS utility to generate current statistics in the DB2 catalog about any DB2 table space, including those in the Tivoli Decision Support for z/OS database.

The sample job, DRLJRUNS (in the DRL182.SDRLCNTL library), calls the DB2 RUNSTATS utility. Figure 37 on page 109 shows DRLJRUNS used to generate statistics for table spaces DRLSSYS1 and DRLSSYS2.

Administering the Tivoli Decision Support for z/OS database

```
//DRLJRUNS JOB (ACCT#),'RUNSTATS'  
//*****  
//* *  
//* Licensed Materials - Property of IBM *  
//* *  
//* 5698-B06 Copyright IBM Corporation 1992, 2015 *  
//* See Copyright Instructions. *  
//* *  
//*****  
//* *  
//* Name: DRLJRUNS *  
//* *  
//* Status: Tivoli Decision Support for zOS 1.8.2 *  
//* *  
//* Function: *  
//* Run the DB2 RUNSTATS utility to update the DB2 catalog *  
//* information about Performance Reporter tables. *  
//* This job only runs RUNSTATS for the table spaces *  
//* DRLSSYS1 and DRLSSYS2. You must add a statement for *  
//* each Performance Reporter table space. *  
//* *  
//* Notes: *  
//* Check the following: *  
//* LIB='db2loadlibrary' DB2 load library *  
//* SYSTEM=DSN DB2 subsystem name *  
//* *  
//* *  
//* CHANGE ACTIVITY: *  
//* CHANGE FLAG TYPE DATE DESCRIPTION *  
//* ----- *  
//* $D0=DCR066, TDS182,01/06/15,ADL(SM): Update TDS Version and *  
//* DB2 dataset names. *  
//* *  
//*****  
//* *  
//UTIL EXEC DSNUPROC,LIB='db2loadlibrary',  
// SYSTEM=DSN,UID='TEMP',UTPROC=''  
//* *  
//DSNUPROC.SYSIN DD *  
RUNSTATS TABLESPACE DRLDB.DRLSSYS1 TABLE INDEX  
RUNSTATS TABLESPACE DRLDB.DRLSSYS2 TABLE INDEX  
/*
```

Figure 37. DRLJRUNS job for generating DB2 statistics

Learn more about the DB2 RUNSTATS utility from the description of its use in the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

Start the RUNSTATS utility from the administration dialog by choosing it from the Utilities menu in the Tables window. After using the RUNSTATS utility, use the administration dialog to see the number of bytes used for data in the product database (described in “Showing the size of a table” on page 167).

Understanding how Tivoli Decision Support for z/OS uses DB2 locking and concurrency

DB2 provides locking and dynamic recovery for the databases it controls. The Tivoli Decision Support for z/OS database is under DB2 control and uses these DB2 mechanisms.

More than one Tivoli Decision Support for z/OS user or function can request access to the data at the same time. The way DB2 maintains data integrity during such times is by locking out data to all processes but one.

Administering the Tivoli Decision Support for z/OS database

Learn more about DB2 locking and how it allows more than one process to work with data concurrently from the discussion of improving concurrency in the *Guide to Reporting*, SH19-6842.

Deadlock or timeout conditions can occur when more than one user works with Tivoli Decision Support for z/OS tables, which causes DB2 to generate messages; for example:

```
DSNT408I  SQLCODE = -911, ERROR:  THE CURRENT UNIT OF WORK HAS BEEN
          ROLLED BACK DUE TO DEADLOCK OR TIMEOUT.  REASON 00C90088,
          TYPE OF RESOURCE 00000100, AND RESOURCE NAME DRLDB
```

Consider the following potential locking scenarios:

- If running more than one collect job at a time, ensure the jobs do not update the same tables.
Although concurrent collects might not update the same data tables, locking can occur for the DRLSYS.DRLLOGDATASETS system table, updated by all collect runs.
- Generating reports while a collect job runs does not usually cause lockouts.
Report queries do not update table information; their access is read-only. However, QMF can hold locks while you display large reports.
- You cannot collect while DB2 utilities such as COPY and REORG are running.
Also, you cannot collect and purge simultaneously.
COPY and REORG lock all tables in the table space on which they operate.
Purge locks the table on which it operates.
- Creating tables (or installing components) locks the entire database.
If some users create many tables, give them a private database. See “Installing multiple Tivoli Decision Support for z/OS systems” on page 49 for more information.

To find out who is locking a resource, use the DB2 COMMANDS option in DB2 to issue this command:

```
-DISPLAY DATABASE(DRLDB) LOCKS LIMIT(100)
```

For more information, refer to the description of monitoring DB2 locking in the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

Maintaining database security

About this task

You control user access to database tables. Although Tivoli Decision Support for z/OS grants read access to the DRLUSER group ID for any components you install, you can grant or revoke authority to tables in the Tivoli Decision Support for z/OS database. See “Administering user access to tables” on page 202 for more information.

Monitoring database access

About this task

To see which end users access which database tables (for example, if you are considering removing tables), use the DB2 trace facility for tracing table accesses. Analyze the trace outside DB2 with another product. IBM DB2 Performance Monitor (DB2PM) can format, print, and interpret DB2 trace data.

Administering the Tivoli Decision Support for z/OS database

Tracing involves a significant amount of overhead and is not something you should do regularly.

For information about DB2 trace facilities, refer to the description of using tools to monitor performance in the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

For information about DB2PM, refer to the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference* and to the *IBM DB2 Performance Monitor: User's Guide*.

Using available tools to work with the Tivoli Decision Support for z/OS database

About this task

IBM and other software suppliers provide a variety of database maintenance tools. Because you have database administrator authority for the Tivoli Decision Support for z/OS database, you can use tools such as DB2I, a part of DB2. With DB2I you can:

- Run SQL statements
- Issue authorized DB2 commands
- Run DB2 utilities
- Work with DB2 objects in your database

Select DB2I from the Other menu of any Tivoli Decision Support for z/OS primary window. You can also type DB2I on the command line of a window.

Figure 38 shows the DB2I Primary Option Menu.

```
COMMAND ==>>>                                DB2I PRIMARY OPTION MENU

Select one of the following DB2 functions and press ENTER.

 1 SPUFI                (Process SQL statements)
 2 DCLGEN               (Generate SQL and source language declarations)
 3 PROGRAM PREPARATION (Prepare a DB2 application program to run)
 4 PRECOMPILE           (Invoke DB2 precompiler)
 5 BIND/REBIND/FREE     (BIND, REBIND, or FREE plans or packages)
 6 RUN                  (RUN an SQL program)
 7 DB2 COMMANDS        (Issue DB2 commands)
 8 UTILITIES           (Invoke DB2 utilities)
 9 CATALOG VISIBILITY  (Invoke catalog dialogs)
 D DB2I DEFAULTS      (Set global parameters)
 X EXIT                (Leave DB2I)

F13=HELP  F14=SPLIT  F15=END    F16=RETURN  F17=RFIND  F18=RCHANGE
F19=UP    F20=DOWN   F21=SWAP   F22=LEFT   F23=RIGHT  F24=RETRIEVE
```

Figure 38. DB2I Primary Option Menu

For more information about DB2I, refer to the description of utility jobs in the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

Administering lookup and control tables

Periodically review the contents of Tivoli Decision Support for z/OS lookup and control tables. See Chapter 12, “Control tables and common tables,” on page 243 for a description of the columns in lookup and control tables that many product feature components use. Lookup tables used exclusively by a Tivoli Decision Support for z/OS feature are described in the documentation for each feature.

Edit each lookup table and control table to implement standards and definitions at your site. “Working with data in tables” on page 164 describes how to edit tables.

Lookup and control tables are particularly important for reporting availability of resources. Discuss availability reporting with your users to determine necessary changes to these tables.

Administering reports

About this task

As a Tivoli Decision Support for z/OS administrator, you have authority to run all frequently requested reports in batch mode and distribute them regularly. You can also create report groups that suit your organization.

Running reports in batch

About this task

You can generate reports using the reporting dialog. For more information, refer to the *Guide to Reporting*, SH19-6842. However, for frequently requested reports, you should set up jobs that produce the reports regularly.

The steps to do this are as follows:

Procedure

1. Specify batch settings for the reports.
2. Define queries and forms suitable for batch reports.
3. Print reports or save them in data sets, using a batch job or the reporting dialog.
4. >Optionally, save the reports for reporting dialog users and regularly replace the saved report data with new data.
5. Optionally, include saved charts in BookMaster® documents.

Specifying batch settings

Use the Set batch option in the Batch pull-down in the reporting dialog to specify the batch settings for a report. Batch settings include output options and other options.

Understanding output options for batch reports: There are two output options for batch reports:

- Print the report:
 - If your installation uses QMF, tabular reports are printed to the DSQPRINT file. Otherwise they are printed to the DRLPRINT file.
 - Graphic reports are printed to the printer specified in the job (or to the default printer defined in the QMF profile, if no printer is specified).

The printer name must be defined in the GDDM nicknames file, allocated to the ADMDEFS ddname. Refer to the *QMF: Planning and Administration Guide for MVS* and the *GDDM User's Guide* for more information about defining GDDM nicknames.

If you do not use QMF, all reports are printed in tabular format. If you require graphic reports, you can print a saved report with GDDM-PGF or other tools.

- Save the report in a data set:
 - Tabular reports are saved in the data set defined by the DRLREP ddname, usually DRL.LOCAL.REPORTS.
 - Graphic reports are saved in the data set defined by the ADMGDF ddname, usually DRL.LOCAL.CHARTS. `idd:page>`
Saved reports serve different purposes:
 - Set up the reporting dialog to use it to look at saved reports.
 - Display the reports in other ways, such as from user-written applications.
 - Include the reports in BookMaster documents.

Defining report queries and forms for batch execution

Although all Tivoli Decision Support for z/OS reports can be run in batch, most of them are not suited for batch because you must supply values for all the variables in the queries and forms.

For example, a typical query looks like this:

```
SELECT column1, column2, ...
FROM table
WHERE DATE >= &FROM_DATE.
      AND DATE <= &TO_DATE.
      AND SYSTEM_ID = &SYSTEM_ID.
```

When displayed from the dialog, Tivoli Decision Support for z/OS prompts you for values for FROM_DATE, TO_DATE, and SYSTEM_ID. To run the report in batch, you must supply the values in the job and you must change them when you want the reports to cover a different period.

You can change the query to require no variables and always cover the last week:

```
SELECT SYSTEM_ID, column1, column2, ...
FROM table
WHERE DATE >= CURRENT DATE - 7 DAYS
```

Refer to the *Guide to Reporting*, SH19-6842 for a description of how to create a query.

If the form used contains variables other than the standard variables REPORT_TITLE, PRODUCT_NAME, and REPORT_ID, you must make sure that these variables are set in the batch reporting job, or modify the form. Refer to the *Guide to Reporting* for a description of how to create and modify forms.

Using job DRLJBATR to run reports in batch

The sample job, DRLJBATR (in the DRL182.SDRLCNTL library), produces all, or a subset, of the reports that have batch settings specified. 114 shows DRLJBATR.

You need to change some parameters in DRLJBATR to your requirements. For a description of those parameters, see Table 7 on page 117.

DRLJBATR job for printing or saving reports in batch (Part 1 of 2)

Administering reports

```

//DRLJBATR JOB (ACCT#),'REPORTS'                                00010000
//*****                                                    00020000
//*                                                            * 00030000
//* Licensed Materials - Property of IBM                      * 00040002
//*                                                            * 00050000
//* 5698-B06 Copyright IBM Corporation 1992, 2009           *
//* See Copyright instructions.                               * 00070002
//*                                                            * 00080000
//*****                                                    00090000
//*                                                            * 00100000
//* Name: DRLJBATR                                           * 00110002
//*                                                            * 00120000
//* Status: Tivoli Decision Support for zOS 1.8.1           * 00130008
//*                                                            * 00140000
//* Function:                                                * 00150002
//* TDS for zOS batch reporting sample job.                 * 00160002
//*                                                            * 00170000
//* This job is used to print and/or save all (or a selected * 00180002
//* subset of) the batch reports.                            * 00190002
//*                                                            * 00200000
//* Reports printed to : DSQPRINT with QMF (tables)          * 00210002
//*                      DRLPRINT w/o QMF (tables)          * 00220002
//*                      printer specified (charts)         * 00230002
//* Reports saved in : DRLREP (tables)                      * 00240002
//*                      ADMGDF (charts)                    * 00250002
//* Messages written to: DRLOUT                              * 00260002
//*                                                            * 00270000
//* The exec DRLEBATR accepts the following parameters:     * 00280002
//*                                                            * 00290000
//* SYSTEM=DB2_system DB2 subsystem name. The default is DSN * 00300002
//* SYSPREFIX=sysprefix Prefix for TDSzOS system            * 00310002
//*                                                            * 00320002
//* PREFIX=prefix Prefix for all other tables. The default * 00330002
//* is DRL.                                                  * 00340002
//* SHOWSQL=YES/NO Show SQL statements (for debugging).     * 00350002
//* YES or NO. The default is NO.                           * 00360002
//* CYCLE=run_cycle Run cycle: DAILY, WEEKLY or MONTHLY.   * 00370002
//* If not specified, all reports are printed.              * 00380002
//* GROUP=report_group Report group. If not specified, all * 00390002
//* reports are printed.                                     * 00400002
//* REPORT=rpt1,rpt2.. Lists the reports to print. If not * 00410002
//* specified, all reports are printed.                      * 00420002
//* PRINTER=prt_name Printer to be used for graphic reports. * 00430002
//* The default printer is defined in the QMF profile.      * 00440002
//*                                                            * 00450002
//* DIALLANG=n Define the application language. PN46029 * 00460002
//* n=1 for English (default)                                * 00470002
//* n=2 for German                                          * 00480002
//* n=3 for Japanese                                       * 00490002
//* QMF=YES/NO Report generation with or PN48405 * 00500002
//* w/o QMF. YES or NO. Default is YES.                     * 00510002
//* GDDM=YES/NO GDDM available for graphic PN48405 * 00520002
//* reports. YES or NO. Default is YES.                     * 00530002
//* DRLMAX=nnnn Max number of result rows from PN48405 * 00540002
//* a query w/o QMF. Default is 5000.                       * 00550002
//* PAGELEN=nn Page length used when printing PN48405 * 00560002
//* tabular reports w/o QMF. Default is 60.                 * 00570002
//* PAGE=PAGE This word is used in the report PN48405 * 00571009
//* footing for page numbering tabular                      * 00572009
//* reports w/o QMF. Default is PAGE                         * 00573009
//* TOTAL=TOTAL This word is used for an across EPDM111 * 00580009
//* summary column header in tabular                        * 00590009
//* reports w/o QMF. Default is TOTAL                       * 00600009
//* DECSEP=PERIOD PERIOD/COMMA. Decimal separator EPDM111 * 00601009
//* setting for tabular reports without QMF.               * 00602009
//* DUALSAVE=xxx Allow graphic reports to be saved PN65801 * 00603015
//* as tabular reports simultaneously.                      * 00604015

```

Administering reports

```

/**          YES/NO (default=NO)                                * 00605015
/** &variable=value      Give a value to a variable used in a   * 00610002
/**                    query or a form. All variables used in   * 00620002
/**                    queries or forms MUST be given a value.  * 00630002
/**                    ' ' = all values for that variable PN65801 * 00630115
/**                    '''''' means the null value.           PN65801 * 00630215
/**                    NB: for variables used with IN operator   *
/**                    '('x') OR (1=1)' = all values           PQ92756 *
/** PRODNAM=Tivoli Decision Supp                               PN46029 * 00640002
/**                    This text is used in the report footing. * 00650002
/**                    The default is TDSzOS                   * 00660002
/**                    Note: If specified, PRODNAM must be the  * 00670002
/**                    last parameter.                          * 00680002
/**                    * 00690002
/** Notes:                                                     * 00700002
/** Before you submit the job, do the following:                * 00710002
/** 1. Check that the data set names are correct. Update 'DRLvrM' * 00720002
/**    to match your HLQ for TDSz data sets.                    * 00730002
/** 2. Change the parameters to DRLEBATR as required.           * 00740002
/** 3. Remove QMF DD-statements if you are not using QMF. PN48405 * 00750002
/**    Search on 'DSQ' to find such occurrences.                PN48405 *
/**    The exception is DSQCFRM, which should be changed PN68060 * 00760000
/**    to DRLUFORM. The dataset name should point to the       * 00760100
/**    user defined forms library.                              * 00760200
/** 4. Change the DB2 load library name according to            * 00270000
/**    the naming convention of your installation.              * 00260000
/**    Default is 'db2loadlibrary'.                             * 00260000
/**                                                            * 00760300
/** CHANGE ACTIVITY:                                           * 00770002
/** 00 1993-05-18 JHS Created                                   * 00780002
/** 01 1993-10-25 JCS Variables DIALLANG and PRODNAM PN46029 * 00790002
/** 02 1993-12-01 LW Reporting without QMF                    PN48405 * 00800002
/** 03 1994-11-15 IW DRLFORM DD card added for QMF form       * 00801013
/**                    and SYSEXEC set to same as SYSPROC PN65801 * 00810015
/** 04 1994-12-12 LW Allow 'dual' save                         PN65801 * 00810115
/** 05 1994-12-15 LW ADMPRNTQ added                           PN65906 * 00810215
/** 06 1995-02-16 PN Comment concerning DRLUFORM for          PN68060 * 00810216
/**                    non-QMF users added above.              * 00810217
/** 07 2004-11-03 RV Sysroute of apar PQ92756                 PQ96265 *
/**                                                            * 00811013
/**                                                            *
/** CHANGE ACTIVITY:                                           *
/** CHANGE FLAG TYPE DATE DESCRIPTION                          *
/** -----*
/** $D0=DCR066, TDS180,01/06/07,ADL(SM): Update TDS Version and *
/**                    DB2 dataset names.                       *
/** $D1=DCR116, TDS181,15/05/09,ADL(RC): Update TDS Version   *
/**                                                            *
/** ***** 00820000
//REPORT EXEC PGM=IKJEFT01 00830000
//* 00840000
//STEPLIB DD DISP=SHR,DSN=DRLvrM.SDRLOAD 00850008
// DD DISP=SHR,DSN=qmfloadlibrary 00860000
// DD DISP=SHR,DSN=db2loadlibrary 00870000
//SYSPROC DD DISP=SHR,DSN=DRLvrM.SDRLEXEC 00880008
// DD DISP=SHR,DSN=qmfclistlibrary 00890000
//SYSEXEC DD DISP=SHR,DSN=DRLvrM.SDRLEXEC 00891013
// DD DISP=SHR,DSN=qmfexeclibrary 00900013
//***** 00910000
//* MESSAGES 00920002
//* 00930000
//DRLOUT DD SYSOUT=* 00940000
//***** 00950000
//* PRINT REPORTS TO EITHER DSQPRINT OR DRLPRINT PN48405 00960002
//* 00970000
//DSQPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330) 00980000
//DRLPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330) 00990000

```

Administering reports

```

//*****
//* SAVE REPORTS IN
//*
//DRLREP DD DISP=SHR,DSN=DRL.LOCAL.REPORTS
//ADMGDF DD DISP=SHR,DSN=DRL.LOCAL.CHARTS
//*****
//* GDDM LIBRARIES
//*
//ADMGGMAP DD DISP=SHR,DSN=ADMGGMAPlibrary
//ADMCFORM DD DISP=SHR,DSN=ADMCFORMlibrary
// DD DISP=SHR,DSN=DRLvrn.SDRLFENU
//ADMSYMBL DD DISP=SHR,DSN=SYS1.GDDMSYM
//ADMDEFS DD DISP=SHR,DSN=SYS1.GDDMNICK
//*ADMPRINT DD DISP=SHR,DSN=ADMPRINT.REQUEST.QUEUE
//DSQCFCRM DD DISP=SHR,DSN=DRLvrn.SDRLFENU
//*****
//* QMF LIBRARIES
//*
//DSQDEBUB DD DUMMY
//DSQDUMP DD DUMMY
//DSQPNL DD DISP=SHR,DSN=QMFDSQPNLlibrary
//DSQSPILL DD DSN=&&SPILL,DISP=(NEW,DELETE),UNIT=SYSDA,
// SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096)
//DSQEDIT DD DSN=&&EDIT,UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
// DCB=(RECFM=FBA,LRECL=79,BLKSIZE=4029)
//DRLFORM DD DSN=&&FORMDS,UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
// DCB=(RECFM=VB,LRECL=255,BLKSIZE=2600),DISP=(NEW,DELETE)
//*****
//* START EXEC DRLEBATR
//*
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%DRLEBATR SYSTEM=DSN SYSPREFIX=DRLSYS PREFIX=DRL -
PRINTER=XXX -
REPORT=XXXXXXXX,YYYYYYYY -
&SYSTEM_ID='SYS1' -
&FROM_DATE='1993-01-01' -
&TO_DATE='1993-04-01' -
DIALLANG=1 -
PRODDNAME=Tivoli Decision Supp
/*
//*****
//* GDDM LIBRARIES
//*
//ADMGGMAP DD DISP=SHR,DSN=ADMGGMAPlibrary
//ADMCFORM DD DISP=SHR,DSN=ADMCFORMlibrary
// DD DISP=SHR,DSN=DRL182.SDRLFENU
//ADMSYMBL DD DISP=SHR,DSN=SYS1.GDDMSYM
//ADMDEFS DD DISP=SHR,DSN=SYS1.GDDMNICK
//*ADMPRINT DD DISP=SHR,DSN=ADMPRINT.REQUEST.QUEUE
//DSQCFCRM DD DISP=SHR,DSN=DRL182.SDRLFENU
//*****
//* QMF LIBRARIES
//*
//DSQDEBUB DD DUMMY
//DSQDUMP DD DUMMY
//DSQPNL DD DISP=SHR,DSN=QMFDSQPNLlibrary
//DSQSPILL DD DSN=&&SPILL,DISP=(NEW,DELETE),UNIT=SYSDA,
// SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096)
//DSQEDIT DD DSN=&&EDIT,UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
// DCB=(RECFM=FBA,LRECL=79,BLKSIZE=4029)
//DRLFORM DD DSN=&&FORMDS,UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
// DCB=(RECFM=VB,LRECL=255,BLKSIZE=2600),DISP=(NEW,DELETE)
//*****
//* START EXEC DRLEBATR
/*

```

```

01000000
01010002
01020000
01030000
01040000
01050000
01060002
01070000
01080000
01090000
01100008
01110000
01120000
01121015
01130008
01140000
01150002
01160000
01170000
01180000
01190000
01200000
01210000
01220000
01230000
01230312
01231010
01240000
01250002
01260000
01270000
01280000
01290000
01300000
01310000
01320000
01330000
01340000
01350002
01360003
01370003
01380000

```

```

//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%DRLEBATR SYSTEM=DSN  SYSPREFIX=DRLSYS  PREFIX=DRL  -
  PRINTER=XXX  -
  REPORT=XXXXXXXX,YYYYYYYY  -
  &SYSTEM_ID='SYS1'  -
  &FROM_DATE='1993-01-01'  -
  &TO_DATE='1993-04-01'  -
  DIALLANG=1  -
  PRODDNAME=Tivoli Decision Supp
/*

```

Using the reporting dialog to run reports in batch

To create reports in batch from the reporting dialog:

1. From the Tivoli Decision Support for z/OS Administration window, select 5, Reports, and press Enter to display the Reports window.
2. Without selecting any reports in the Tivoli Decision Support for z/OS Reports window, select the Invoke batch option from the Batch pull-down. The Batch Reports Selection window is displayed.
3. Type required information, such as whether to run daily, weekly, or monthly reports, and press Enter. If any of the reports contain variables, the Batch Reports Data Selection window is displayed.
4. Specify values to select the data to be reported, and press Enter to display the job.
5. Edit the job, specifying the parameters described in “Parameters for batch reporting.” Then type SUBMIT on the command line, and press Enter.
Tivoli Decision Support for z/OS submits your job to run in background.
6. Press F3 to return to the Reports window.

Refer to the *Guide to Reporting* for more information about running reports in batch.

Parameters for batch reporting

Table 7. Parameters for batch reporting

Parameter	Value to set	Explanation	Default value	Your value
SYSTEM	DB2 subsystem name (UPPERCASE)	<p>The DB2 subsystem where Tivoli Decision Support for z/OS resides.</p> <p>This required parameter can be 4 alphanumeric characters. The first character must be alphabetic.</p> <p>The default value is DSN. If the value in this field is something other than DSN, it was changed during installation to name the correct DB2 subsystem.</p> <p>Do not change the value to name another DB2 subsystem to which you might have access. Tivoli Decision Support for z/OS must use the DB2 subsystem that contains its system, control, and data tables.</p>	DSN	

Administering reports

Table 7. Parameters for batch reporting (continued)

Parameter	Value to set	Explanation	Default value	Your value
SYSPREFIX	Prefix for system tables (UPPERCASE)	<p>The prefix of all Tivoli Decision Support for z/OS system and control DB2 tables. The value of this field depends upon your naming conventions and is determined during installation.</p> <p>This required parameter can be 8 alphanumeric characters. The first character must be alphabetic.</p> <p>The default is DRLSYS. If the value is something other than DRLSYS, it was changed during installation.</p> <p>Do not change the value; Tivoli Decision Support for z/OS uses this value to access its system tables.</p>	DRLSYS	
PREFIX	Prefix for all other tables (UPPERCASE)	<p>The prefix of Tivoli Decision Support for z/OS data tables in the DB2 database.</p> <p>Valid values are determined at installation.</p> <p>This required parameter can be 8 alphanumeric characters. The first character must be alphabetic.</p> <p>The default is DRL. If the value is something other than DRL, it was changed during installation.</p>	DRL	
SHOWSQL	YES or NO (UPPERCASE)	Here you specify if SQL statements should be shown (for debugging purposes).	NO	
CYCLE	DAILY, WEEKLY or MONTHLY (UPPERCASE)	The run cycle for reports. If you do not specify daily, weekly, or monthly, all reports are printed.	All reports	
GROUP	A report group ID (UPPERCASE)	Here you can specify the ID of a report group. If you do not specify a group, all reports are printed.	All reports	
REPORT	One or more report IDs (UPPERCASE)	Here you can specify one or more reports to be printed. If you do not specify any reports, all reports are printed.	All reports	
PRINTER	Default printer name (UPPERCASE)	<p>The GDDM nickname of a printer to use for printing graphic reports. The printer should be capable of printing GDDM-based graphics.</p> <p>The printer name must be defined in the GDDM nicknames file, allocated to the ADMDEFS ddname. Refer to the <i>QMF: Reference</i> and <i>GDDM User's Guide</i> for more information about defining GDDM nicknames.</p> <p>This parameter cannot be used if QMF=NO.</p>	As defined in the QMF profile	
DIALLANG	1. English 2. Japanese	With this parameter, you specify the language to be used.	1=English	
QMF	YES or NO (UPPERCASE)	With this parameter, you specify whether your installation uses QMF or not.	YES	
GDDM	YES or NO (UPPERCASE)	With this parameter, you specify if your installation uses GDDM.	YES	
DRLMAX	nnnn	If your installation does not use QMF, you use this parameter to specify the maximum number of result rows from a query.	5000	

Table 7. Parameters for batch reporting (continued)

Parameter	Value to set	Explanation	Default value	Your value
PAGELEN	nn	If your installation does not use QMF, you use this parameter to specify the page length when printing tabular reports.	60	
PAGE	The word for page (Mixed case)	If your installation does not use QMF, the word you specify here is inserted before the page number for tabular reports. You can type the word in mixed case, for example, Page.	PAGE	
TOTAL	The word for total (Mixed case)	If your installation does not use QMF, the word you specify here is used as column heading for across summary columns in tabular reports. You can type the word in mixed case, for example, Total.	TOTAL	
DECSEP	Period or comma	If your installation does not use QMF, you use this parameter to specify the decimal separator to be used in tabular reports. If you use a comma as a decimal separator, a period is used as thousands separator, if applicable.	PERIOD	
DUALSAVE	YES or NO (UPPERCASE)	Allow graphic reports to be saved as tabular reports simultaneously.	NO	
&variable	A value	This parameter gives a value to a variable used in a query or form. All variables used in queries or forms must be given a value.		
PRODNAME	Tivoli Decision Support for z/OS Report (Mixed case)	This text is used in the report footer. If specified, PRODNAME must be the last parameter.	Tivoli Decision Support for z/OS Report	

Saving reports for reporting dialog users

You can save report data from a reporting job like DRLJBATR. Creating reports for batch preprocessing and then saving them for end users means:

- Users need not access the Tivoli Decision Support for z/OS database if they have access to current reports instead.
- Users need not take the time to run reports.
- Users have the data they need to begin analysis immediately.

To preprocess reports for dialog users:

1. Define the batch report as described in “Specifying batch settings” on page 112.
2. Select the batch report and select 4, Save report data, from the Reports pull-down. The Saved Report Definition window is displayed. Refer to the *Guide to Reporting* for information about defining saved reports in the Saved Report Definition window.
3. After completing all fields in the Saved Report Definition window, press Enter. The report is run and saved in the specified member.
4. Add the saved report to a report group, such as *Monthly Management Reports*, to let users display relevant reports easily.
Refer to the *Guide to Reporting* for information about adding a report to a report group.

After you complete the steps above, you can run the batch report periodically (using the DRLJBATR job) to replace the saved report member with up-to-date information.

Administering reports

Including saved charts in BookMaster documents

Tivoli Decision Support for z/OS produces graphic reports in ADMGDF format. It saves them to the data set identified by the job's ADMGDF ddname or the Saved chart data set field of the Dialog Parameters window. To include charts in a BookMaster document, convert them to page segments (PSEGs).

The GDDM-PGF utility, ADMUCDSO, can perform the conversion. Figure 39 shows a sample job for producing a page segment. Refer to the *GDDM-PGF Programming Reference* for a complete description of the utility.

```
//job card
//TSO      EXEC PGM=IKJEFT01
//ADMGDF   DD DISP=SHR,DSN=DRL.LOCAL.CHARTS           IN:  ADMGDF
//ADMIMAGE DD DISP=SHR,DSN=xxx.xxx.PSEG3820(SAMPLE01) OUT: PSEG3820
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
CALL 'SYS1.LINKLIB(ADMUCDSO)' -
'SAMPLE01 GDF 99 4 IMG240 (5 1 6 0 7 10 8 65 45 0 9 1) (ADMIMAGE)'
/*
```

Figure 39. Converting saved graphic report data to a page segment

To use the SAMPLE01 page segment in a BookMaster document that Document Composition Facility (DCF) can format, you use an artwork tag (Figure 40):

```
:h1.Sample Report 1
:p.This surface chart shows the CPU time consumed by different projects.
It gives an hourly profile for an average day.
:artwork name=sample01.
```

Figure 40. Using reports in BookMaster documents

QMF batch reporting

Batch reporting can also be performed with QMF only, without using Tivoli Decision Support for z/OS functions. A QMF job can simply execute a QMF procedure that contains QMF commands (Figure 41).

```
RUN QUERY1 (FORM=FORM1
PRINT REPORT
RUN QUERY2 (FORM=FORM2
PRINT REPORT (PRINTER=LOCAL1
```

Figure 41. Using QMF to report in batch

These books contain more information about using QMF in this way:

- *QMF Advanced User's Guide*
- *QMF Reference*

Creating report groups

About this task

Tivoli Decision Support for z/OS reports are grouped by component within each feature. Placing more commonly requested reports in new report groups can make retrieving them easier. Creating report groups for users with special requirements, such as managers, also makes reporting more effective.

Refer to the *Guide to Reporting* for information about creating report groups.

Administering problem records

About this task

The update definitions of some Tivoli Decision Support for z/OS components update the common table, EXCEPTION_T, with data about system exceptions that require attention. Review this information and use the product interface for adding selected exceptions to the Tivoli Information Management for z/OS database.

You can review exceptions only through the administration dialog. You can generate problem records with either the dialog or a job.

Reviewing exceptions and generating problem records

About this task

To review exceptions and generate problem records:

Procedure

1. Select 2, Generate problem records, from the Utilities pull-down of the Tivoli Decision Support for z/OS Administration window and press Enter.
The Exception Selection window is displayed.
2. Type 2, No, in the Problems only field to list all exception records.>

Note: The default update definitions do not classify exceptions as problems. You can modify them to set the problem flag (column PROBLEM_FLAG='Y' in the EXCEPTION_T table).

3. Type 1, Yes, in the Not generated only field to select exception records that have not yet been generated as problem records in the Tivoli Information Management for z/OS database.
4. Select values for other required fields in the window.
Use the fields to restrict the number of exceptions in the list of exceptions.
Use F4 (Prompt) to see a selection list for any field in the Exception Selection window.
5. Press Enter to see the list of exceptions.
The Exception List window is displayed.
6. Select an exception and press Enter
The Generate Record window is displayed, showing the exception record in detail.
7. If the exception record is one you want to add to the Tivoli Information Management for z/OS database, press Enter.
Tivoli Decision Support for z/OS generates the problem record.

Generating problem records in batch

About this task

Although the sample job, DRLJEXCE (in the DRL182.SDRLCNTL library) does not let you review exception records, it generates problem records in the Tivoli Decision Support for z/OS database only from EXCEPTION_T records defined as problems.

Administering problem records

Note: You must customize the product update definitions that add records to EXCEPTION_T to set the problem flag column.

DRLJEXCE job for generating problem records

```
//DRLJEXCE JOB (ACCT#),'EXCEPTION REPORTING'                                00010003
//*****                                                                    00020000
//*                                                                           * 00030000
//* LICENSED MATERIALS - PROPERTY OF IBM                                  * 00040004
//*                                                                           * 00050000
//* 5698-B06 COPYRIGHT IBM CORPORATION 1993, 2015                       * 00060004
//* SEE COPYRIGHT INSTRUCTIONS.                                          * 00070004
//*                                                                           * 00080000
//*****                                                                    00090000
//*                                                                           * 00100000
//* NAME:      DRLJEXCE                                                  * 00110004
//*                                                                           * 00120000
//* STATUS:    Tivoli Decision Support for zOS 1.8.2                    * 00130004
//*                                                                           * 00140000
//* FUNCTION:  EXCEPTION REPORTING.                                       * 00150004
//*           PROBLEM RECORDS ARE GENERATED BY TIVOLI SERVICE DESK      * 00160004
//*           FOR ALL RECORDS IN THE EXCEPTION TABLE (EXCEPTION_T),    * 00160004
//*           WHERE                                                       * 00170004
//*           A) THE PROBLEM_FLAG COLUMN INDICATES THAT THIS RECORD      * 00180004
//*              IS A PROBLEM RECORD (PROBLEM_FLAG='Y')                  * 00190004
//*           B) AND THE DATE_GENERATED COLUMN INDICATES THAT THE        * 00200004
//*              TIVOLI SERVICE DESK DATABASE HAS NOT BEEN UPDATED       * 00210004
//*              WITH THIS RECORD (DATE_GENERATED IS NULL).              * 00210004
//*                                                                           * 00230000
//* INPUT PARAMETERS:                                                    * 00240004
//*   SYSTEM=DB2-SUBSYSTEM      DB2 SUBSYSTEM   (DEFAULT=DSN)           * 00250004
//*   PREFIX=PREFIX            TABLE PREFIX   (DEFAULT=DRL)            * 00260004
//*   MODE=BATCH               BATCH/ONLINE   (DEFAULT=BATCH)          * 00270004
//*   APPLID=XXXXXX           APPLICATION ID   (DEFAULT=SYSUID)         * 00280004
//*   SESSMBR=XXXXXX         SESSION MEMBER   (DEFAULT=BLGSES00)       * 00290004
//*   PRIVCLASS=XXXXXX       PRIVILEGE CLASS  (DEFAULT=MASTER)        * 00300004
//*                                                                           * 00310000
//* OUTPUT:  - PROBLEM RECORD(S) CREATED IN TIVOLI SERVICE DESK.        * 00320004
//*           - TABLE EXCEPTION_T UPDATED WITH PROBLEM NUMBER          * 00330004
//*           AND DATE GENERATED.                                        * 00340004
//*           - RESULT FILE WRITTEN TO FILE DEFINED BY DRLOUT DD.       * 00350004
//*                                                                           * 00360000
//* NOTES:   BEFORE YOU SUBMIT THIS JOB, DO THE FOLLOWING:              * 00370004
//*   1. ENSURE THAT YOU (OR THE VALUE SPECIFIED BY APPLID) ARE         * 00380004
//*      REGISTERED AS A VALID APPLICATION ID IN TIVOLI SERVICE DESK.    * 00390004
//*   2. CHECK THAT THE DATASET NAMES ARE CORRECT.                     * 00400004
//*   3. CHANGE THE PARAMETERS TO DRLEREGE AS REQUIRED.                  * 00410004
//*   4. Change the DB2 load library name according to                   * 00270000
//*      the naming convention of your installation.                     * 00260000
//*      Default is 'db2loadlibrary'.                                    * 00260000
//*                                                                           * 00420000
//*                                                                           *
//* CHANGE ACTIVITY:                                                    *
//* CHANGE FLAG  TYPE      DATE      DESCRIPTION                          *
//* -----*
//* $D0=DCR066, TDS181,01/06/13,ADL(SM): Update TDS Version and        *
//*                                           DB2 dataset names.          *
//* $D1=DCR116, TDS182,15/05/15,ADL(RC): Update TDS Version           *
//*                                                                           *
//*****                                                                    00430000
//*                                                                           00440000
//EPDMEXCE EXEC PGM=IKJEFT01,DYNAMNBR=25                                00450000
//STEPLIB DD DISP=SHR,DSN=TSDB.SBLMMOD1                                00460000
//          DD DISP=SHR,DSN=DRLvrms.SDRLLLOAD                          00470002
//          DD DISP=SHR,DSN=db2loadlibrary                              00480000
//SYSPROC DD DISP=SHR,DSN=DRLvrms.SDRLEXEC                             00490002
//*-----*                                                                00500000
```

Administering problem records

```
//* TIVOLI SERVICE DESK LIBRARIES * 00510004
/*-----* 00520000
//BLGSD DD DISP=SHR,DSN=TSO.SDDS 00530000
//BLGSI DD DISP=SHR,DSN=TSO.SDIDS 00540000
//BLGSL DD DISP=SHR,DSN=TSO.SDIDS 00550000
//BLGPNL0 DD DISP=SHR,DSN=TSO.IBMPNLS 00560000
//BLGPNL1 DD DISP=SHR,DSN=TSO.RPANEL1 00570000
//BLMFMT DD DISP=SHR,DSN=TSO.BLMFMT 00580000
//ISPLLIB DD DISP=SHR,DSN=TSO.SBLMOD1 00590000
/*-----* 00600000
//DRLOUT DD SYSOUT=* 00610000
//SYSPRINT DD SYSOUT=* 00620000
//SYSTSPRT DD SYSOUT=* 00630000
//SYSTSIN DD * 00640000
%DRLEREGE SYSTEM=DSN PREFIX=DRL MODE=BATCH 00650000
/* 00660000
```

Administering problem records

Chapter 7. Working with components

This section describes how to use the administration dialog to work with components. After reading this section, you should be familiar with these tasks:

- “Installing a component” on page 126
- “Uninstalling a component” on page 134
- “Working with a component definition” on page 136

In Tivoli Decision Support for z/OS, a *component* refers to a logical group of objects used to collect log data from a specific source, to update the product database using that data, and to create reports from data in the database. Grouping objects into a component enables you to:

- Install or remove (uninstall) a set of related objects as a package
- View and work with a set of related objects

Each Tivoli Decision Support for z/OS component can include:

- Log collector definitions for:
 - Log types
 - Log procedures
 - Record types in log data sets
 - Record procedures
 - Update definitions
- SQL statements that define these DB2 objects for the component:
 - Table spaces
 - Tables
 - Lookup tables
 - Views
- Report definitions for the component:
 - Report groups
 - Reports

| Each Tivoli Decision Support for z/OS Key Performance Metrics (KPM) component
| also includes table space profiles. Refer to the following section on working with
| table space profiles before installing any of the Key Performance Metrics
| components.

Definition members in product libraries contain component object definitions. You can use the administration dialog to examine statements in these definitions. For an explanation of the statements, see the *Language Guide and Reference*, SH19-6817.

You can use the administration dialog to work with components. From the Administration window (see Figure 3 on page 10), select 2, Components, and press Enter.

The Components window is displayed.

Installing and uninstalling a component

The Components window lists the components available for Tivoli Decision Support for z/OS installation on your system. When you install a component, Tivoli Decision Support for z/OS executes definitions in the component to define all its objects. Then you can use the component to collect, store, and create reports on log data that it supports.

If you no longer need a component, you can use the administration dialog to uninstall it. When you *uninstall* a component, the product deletes from its system tables all objects in that component that are not used by any other installed component. It also deletes all of the component's DB2 objects, including tables and table spaces. The data sets that contain object definition statements are still available, so you can reinstall the component if necessary. The component still appears in the list in the Components window. "Uninstalling a component" on page 134 describes this procedure.

Installing a component

Procedure

1. Refer to these books to plan the tasks you must perform to complete the installation:

Feature

Book name

AS/400 Performance

AS/400 System Performance Feature Guide and Reference

CICS Performance

CICS Performance Feature Guide and Reference

Distributed Systems Performance

Distributed Systems Performance Feature Guide and Reference

IMS Performance

IMS Performance Feature Guide and Reference

Network Performance

Network Performance Feature Installation and Administration

System Performance

System Performance Feature Guide

Resource Accounting

Resource Accounting for z/OS

2. If you want to review DB2 parameters before installing a component, select the component in the Components window, and select Space, as shown in Figure 42 on page 127.

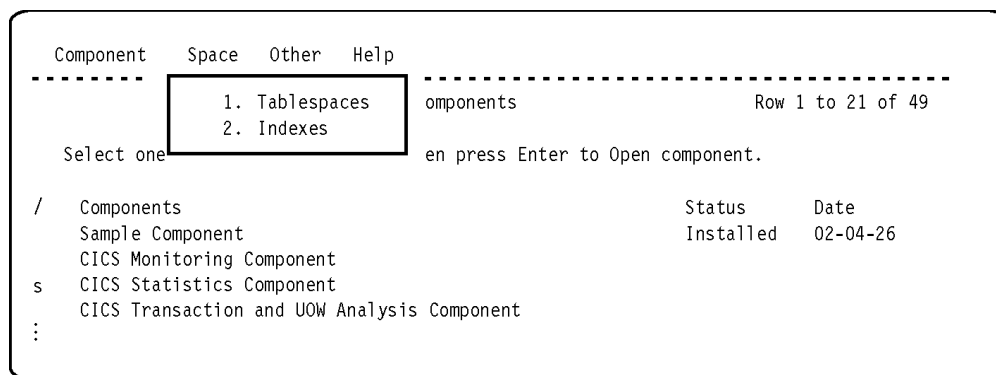


Figure 42. Space pull-down

You can use this pull-down to review and change DB2 space parameters such as:

- Buffer pool
- Compression
- Erase on deletion
- Free space
- Lock size
- Number of partitions, for a partitioned space
- Number of subpages, for an indexspace
- Primary and secondary space
- Segment size
- Type of space
- VSAM data set password

These parameters can affect the performance of your system. If you are unsure how these parameters affect your system, you are recommended to use the defaults provided with the product. If you are unsure about the meaning of a field, press F1 to get help. You should also refer to the CREATE INDEX and CREATE TABLESPACE command descriptions in *DB2 Universal Database for OS/390 and z/OS: SQL Reference*.

Tivoli Decision Support for z/OS saves the changed definitions in your local definitions library. When you save a changed definition, it tells you where it is saving it, and prompts you for a confirmation before overwriting a member with the same name.

3. From the **Components** window, select the component to install and press F6 (Install).

If the component you selected contains subcomponents, the Component Parts window is displayed. Either select the subcomponents to install or press F12 to install only those objects that are not in a subcomponent. (Tivoli Decision Support for z/OS might install some common definitions for the component even though you do not select any of the parts to install.) The **Installation Options** window is displayed.

Installing and uninstalling a component

```
Component  Space  Other  Help
-----
Installation Options                                TO 30 OF 48
S  Select one of the following.  Then press Enter.
/
/  1.  Online
/  2.  Batch                                         Date
-  F1=Help    F2=Split    F6=Objects    F9=Swap    F12=Cancel
/
-  RACF Component
-  Sample Component
-  Storage Management Component
-  VM Accounting Component
-  VM Performance Component
***** BOTTOM OF DATA *****

Command ==>
F1=Help    F2=Split    F3=Exit    F5=New    F6=Install    F7=Bkwd
F8=Fwd     F9=Swap     F10=Actions F12=Cancel
```

Figure 43. Installation Options window

4. From the **Installation Options** window, decide whether to install the component online or in batch mode.
From the **Installation Options** window, you can press F6 (Objects) to see a list of objects in the component. This gives you some idea of its size.
Batch installation leaves an audit trail of what it has done in its spooled output.
Installing a component locks write access to the database, whether you choose online or batch installation. While batch installation occurs, you can use Tivoli Decision Support for z/OS to do anything but update a table in the Tivoli Decision Support for z/OS database. You can also use your terminal to perform any ISPF or TSO task.
5. Select 1 (online) or 2 (batch) and press Enter.
If installing the component online, see the next section, “Installing the component online.”
If installing the component in batch mode, see “Installing the component in batch mode” on page 130.

Installing the component online

About this task

Tivoli Decision Support for z/OS runs the SQL, log collector, and report definition statements to create the objects in the component. The resulting messages are displayed in a browse window:

Procedure

1. If the return code is greater than 0, investigate the messages. For example, the following message indicates a problem accessing the database. DB2 messages are described in *DB2 Universal Database for OS/390 and z/OS: Messages*. If you get this message, you must reinstall the component:

```
DSNT408I  SQLCODE = -911, ERROR:  THE CURRENT UNIT OF WORK HAS
        BEEN ROLLED BACK DUE TO DEADLOCK OR TIMEOUT.  REASON
        00C9008E, TYPE OF RESOURCE 00000100, AND RESOURCE
        NAME DRLDB
```


Installing and uninstalling a component

Correct any error conditions that the product discovers, and install the component again. If the return code is 8 or lower, the status of the component status is **Installed**.

If there are no DB2 messages, *userid.DRLOUT* can look like Figure 44.

```
DB2 Messages
SQL statements executed successfully
-----
Line      Log Collector Messages
-----
      93
DRL0125I The record SMF_080 is defined.
      96
DRL0130I The comment is stored for the record SMF_080.
     1007
DRL0201I The update RACFCOMMAND_80 is defined.
     1014
DRL0403I The purge condition for DRL      .RACF_COMMAND_T is added.
     1138
DRL0201I The update RACFLOGON_80 is defined.
     1145
DRL0403I The purge condition for DRL      .RACF_LOGON_T is added.
     1293
DRL0201I The update RACFOPERATION_80 is defined.
     1300
DRL0403I The purge condition for DRL      .RACF_OPERATION_T is added.
     1466
DRL0201I The update RACFRESOURCE_80 is defined.
     1473
DRL0403I The purge condition for DRL      .RACF_RESOURCE_T is added.
-----
Line      Report Definition Messages
-----
     1503
DRL3001I The group RACF is defined.
     1511
DRL3001I The report RACF01 is defined.
     1519
DRL3001I The report RACF02 is defined.
     1527
DRL3001I The report RACF03 is defined.
     1535
DRL3001I The report RACF04 is defined.
     1543
DRL3001I The report RACF05 is defined.
     1551
DRL3001I The report RACF06 is defined.
     1559
DRL3001I The report RACF07 is defined.
-----
```

Figure 44. Sample log collector messages

2. When you finish browsing the output data set, press F3 (Exit).

If the component has lookup tables, the **Lookup Tables** window is displayed (“Installing the component online” on page 128).

Installing and uninstalling a component

```
Component  Space  Other  Help
-----
Lookup Tables                                ROW 1 TO 3 OF 3

Select a lookup table.  Then press Enter to Edit the table in ISPF Edit
mode.

/  Lookup table
-  RACF_EVENT_CODE
-  RACF_RES_OWNER
-  RACF_USER_OWNER
***** BOTTOM OF DATA *****

Command ==>
F1=Help      F2=Split    F5=QMF add  F6=QMF chg  F7=Bkwd     F8=Fwd
F9=Swap      F12=Cancel

-  VM Accounting Component
-  VM Performance Component

Command ==>
F1=Help      F2=Split    F3=Exit     F5=New       F6=Install   F7=Bkwd
F8=Fwd       F9=Swap     F10=Actions F12=Cancel
```

Figure 45. Lookup Tables window

Refer to the appropriate feature book (shown in “Installing a component” on page 126) for a description of its component lookup tables and how you must edit them.

- To edit a lookup table using ISPF edit, select a table, and press Enter.
Tivoli Decision Support for z/OS accesses the ISPF editor where you can edit the lookup table as described in “Editing the contents of a table” on page 165. If you have QMF installed, you can use the QMF table editor to edit tables wider than 255 characters. If the table has more rows than the value you set for the SQLMAX value field in the **Dialog Parameters** window, Tivoli Decision Support for z/OS prompts you to temporarily override the default for this edit session. To edit a lookup table using the QMF table editor in add mode, press F5 (QMF add). To edit a lookup table using the QMF table editor in change mode, press F6 (QMF chg). “Editing the contents of a table” on page 165 also describes using QMF to edit tables.
- After you make any necessary changes to a lookup table, press F3 (Exit) to save your changes.
Tivoli Decision Support for z/OS returns to the **Lookup Tables** window.
- Edit any other lookup tables that the component requires.
When you finish, the installation is complete.
- Press F12 (Cancel).
Tivoli Decision Support for z/OS returns to the **Components window**. The product has changed the Status field for the component to read Installed.
- Press F3 (Exit). The product returns to the **Administration window**.

Installing the component in batch mode

About this task

Tivoli Decision Support for z/OS builds a batch job to run the SQL, log collector, and report definition statements to create the objects in the component. It then initiates an ISPF edit session. You may have to edit the JCL, for example, to change

the job card. Figure 46 shows a job in an ISPF edit session.

```

EDIT ---- XLLOYDA.SPFTEMP2.CNTL -----COLUMNS 001 072

***** ***** TOP OF DATA *****
000001 //XLLOYDAA JOB (ACCOUNT),'NAME'
000002 //*
000003 //*
000004 //*
000005 //RUNLOG EXEC PGM=DRLPLC,
000006 // PARM=('SYSTEM=DSN SYSPREFIX=DRLSYS &PREFIX=DRL',
000007 // '&DATABASE=DRLDB &STOGROUP=DRLSG')
000008 //STEPLIB DD DISP=SHR,DSN=DRLxxx.SDRLLOAD
000009 // DD DISP=SHR,DSN=DSNxxx.SDSNLOAD
000010 //DRLLOG DD DUMMY
000011 //DRLOUT DD DSNNAME=&TEMP,UNIT=SYSDA,
000012 // DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
000013 // SPACE=(CYL,(10,2)),DISP=(NEW,PASS)
000014 //DRLDUMP DD SYSOUT=*,DCB=BLKSIZE=6160
000015 //DRLIN DD *
000016 SQL SET CURRENT SQLID='DRL';
000017 SET USERS='DRLUSER';
000018 // DD DSN=DRLxxx.SDRLDEFS(DRLRS080),DISP=SHR
COMMAND ==> submit SCROLL ==> 0020
F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 46. Editing an installation job

After editing the job:

Procedure

1. Type SUBMIT on the command line and press Enter.
2. Press F3 after submitting the job.
Tivoli Decision Support for z/OS returns to the **Components** window. The Status field shows Batch which does not mean that the job completed, or that it completed successfully. The installation job changes the value to Installed at its successful completion.
3. When the job completes, use a tool such as the Spool Display and Search Facility (SDSF) to look at the job spool.
4. Review messages for errors as described in step 1 on page 128.
5. Exit SDSF (or whatever tool you are using to review the job spool).
6. Exit the Components window.
7. Refer to the book for the appropriate feature for a description of the component lookup tables you must edit.
8. Select 4, Tables, from the Administration window.
The Tables window is displayed.
9. Select 2, Some, from the View pull-down.
The Select Table window is displayed (Figure 47 on page 132).

Installing and uninstalling a component

```

Table Maintenance Utilities Edit View Other Help
-----
Tables ROW 1 TO 13 OF 393
S
/ Select Table
- Type in the selection criteria. Then press Enter to display.
-
- Component . RACF Component _____ +
*
Name . . . . _____
Prefix . . . . _____
Type . . . . 2 1. Table
                2. Lookup
                3. View
F1=Help      F2=Split      F4=Prompt      F9=Swap      F12=Cancel

Command ==>
F1=Help      F2=Split      F3=Exit      F5=Updates      F6=PurCond      F7=Bkwd
F8=Fwd       F9=Swap       F10=Actions   F11=Display     F12=Cancel

```

Figure 47. Select Table window

10. Type the values as shown in Figure 47, and press Enter.
The Tables window is displayed (Figure 48), showing the component's lookup tables only.

```

Table Maintenance Utilities Edit View Other Help
-----
Tables ROW 1 TO 3 OF 3
Select one or more tables. Then press Enter to Open table definition.

/ Tables Prefix Type
- RACF_EVENT_CODE DRL TABLE
- RACF_RES_OWNER DRL TABLE
/ RACF_USER_OWNER DRL TABLE
***** BOTTOM OF DATA *****

Objects of type Tables meeting the selection criteria are listed.
Command ==>
F1=Help      F2=Split      F3=Exit      F5=Updates      F6=PurCond      F7=Bkwd
F8=Fwd       F9=Swap       F10=Actions   F11=Display     F12=Cancel

```

Figure 48. Tables window - showing component's lookup tables

11. Select a table to edit, but do not press Enter.
12. Select an edit option from the Edit pull-down and press Enter.
If you have QMF installed, you can use the QMF table editor to edit tables wider than 255 characters. See "Editing the contents of a table" on page 165.
13. Press F3 (Exit) when you finish selecting and editing lookup tables.
Tivoli Decision Support for z/OS returns to the Administration window.

Installing and uninstalling a component

When the installation of a component ends with an error or warning, and RC=4 or RC=8, this does not necessarily indicate a problem. The following table shows when you can ignore these messages and return codes.

Message	Return Code	Explanation
SQLCODE=-204 name IS AN UNDEFINED NAME	RC=8	You can ignore this message and return code only if it is caused by an SQL ALTER statement that attempts to add a column to a table that has not yet been created.
SQLCODE=+562 A GRANT OF A PRIVILEGE WAS IGNORED BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR	RC=4	You can always ignore this message and return code.
SQLCODE=-601 THE NAME OF THE OBJECT TO BE CREATED IS IDENTICAL TO THE EXISTING NAME name OF THE OBJECT TYPE objecttype	RC=8	You can always ignore this message and return code.
SQLCODE=-612 column name IS A DUPLICATE COLUMN NAME	RC=8	You can always ignore this message and return code.

Testing the component to verify its proper installation Procedure

1. Collect data from a log data set and review any messages, as described in “Using collect messages” on page 90.

Note: Depending on the component you installed, you might not be able to collect its log data in an online collect. Refer to “Collecting data from a log into DB2 tables” on page 146 for more information.

2. Display a table to ensure that it exists and that it contains the correct information as described in the book for the appropriate feature:

Feature name

Book name

AS/400 Performance

AS/400 System Performance Feature Guide and Reference

CICS Performance

CICS Performance Feature Guide and Reference
CICS Performance Feature Guide and Reference

Distributed Systems Performance

Distributed Systems Performance Feature Guide and Reference, SH19-4018

IMS Performance

IMS Performance Feature Guide and Reference, SH19-6825

Network Performance

Network Performance Feature Reference
Network Performance Feature Reference

System Performance

System Performance Feature Reference

For Resource Accounting, see the *Resource Accounting for z/OS, SH19-4495* book.

3. Display a report to ensure it is correctly installed.

Uninstalling a component

About this task

To uninstall a component:

Procedure

1. From the Components window, select the component you want to uninstall. From the Component pull-down, select the Uninstall option.

If the component you selected contains subcomponents, the Component Parts window is displayed. Either select the parts to uninstall or press F12 to cancel. A confirmation window is displayed.

2. Press Enter to confirm the uninstallation.

Tivoli Decision Support for z/OS deletes from its system tables any component definitions not used by other components. It also deletes all DB2 objects of the component or selected subcomponents, including any tables and table spaces. The component remains in the list of components, but with its Status field cleared. If the component contains subcomponents, they remain in the list of subcomponents but with their Status field cleared.

Note: If a component (or subcomponent) including a common object is uninstalled, the common object is not dropped, unless it is the only installed component (or subcomponent) that includes the common object. When a component or subcomponent is uninstalled, all its data tables are dropped and their contents lost.

Working with table space profiles

Tivoli Decision Support for z/OS introduces new functionality to allow you to easily create table spaces, partitioning on tables, and indexes on tables through the use of table space profiles. These profiles are defined in the DRLTKEYS member in your SDRLEDFS library. Two system tables named GENERATE_PROFILES and GENERATE_KEYS are created and loaded from the DRLTKEYS definition file at the time the TDS system tables are created. These two tables will include all the parameters required to create the Key Performance Metrics (KPM) component table spaces, tables, and indexes.

Default profiles are provided which contain default values for all table spaces, tables, and indexes associated with the profile. Parameters which can be customized in a profile include the partitioning method (partition by growth, partition by range, or non-partitioned), primary, and secondary quantities, storage group names, lock sizes, maximum number of partitions, and most other DB2 table space parameters.

If you wish to use the same parameters for the creation of all your table spaces, tables, and indexes across all the Key Performance Metrics components, only one profile is required.

For more detailed information on the GENERATE_PROFILES and GENERATE_KEYS system tables, refer to Chapter 11, "System tables and views," on page 225

Reviewing Key Performance Metrics table space profiles prior to installation

When you have created your system tables, review the parameter values in the GENERATE_PROFILES and GENERATE_KEYS system tables using the Tivoli Decision Support for z/OS table edit facility. Modify parameters such as PRIQTY and SECQTY accordingly. Note that the GENERATE_KEYS system table only needs to be modified if you are using the partition by range partitioning method.

For the Key Performance Metrics components, the default table space type is set to partition by GROWTH. This can be seen in the TABLESPACE_TYPE column of the GENERATE_PROFILE table where the PROFILE name is SMF or IMS and the COMPONENT_ID and SUBCOMPONENT_ID is set to% (meaning all components and subcomponents).

Refer to Chapter 11, “System tables and views,” on page 225 for information on each column within the GENERATE_PROFILES and GENERATE_KEYS system tables.

Creating storage groups when partitioning by range

If you modify your table space profile to partition by range, you need to run the DRLJDBIP job. The DRLJDBIP job creates additional storage groups that are used in the partitioned table spaces of the KPM components where partition by range was set. Be sure to create one storage group per partition that you identified in the GENERATE_KEYS table.

To run DRLJDBIP:

1. Copy member DRLJDBIP in the DRL182.SDRLCNTL library to the &HLQ.CNTL library.
2. Modify the job statement to run your job.
3. Customize the job for your site. Follow the instructions in the job prolog.
4. Submit the job.

Note: A person with DB2 SYSADMIN authority (or someone who has access to the DB2 catalog) must submit the job.

Reviewing the GENERATE statements for table spaces, tables, and indexes

The Key Performance Metrics components make use of the table space profiling by using GENERATE statements when creating table spaces, tables, and indexes. For example:

```
GENERATE TABLESPACE DRLSKD01
        PROFILE 'SMF';
```

Refer to the *Tivoli Decision Support for z/OS Language Guide and Reference* for the syntax and additional information on using the GENERATE statements. Each GENERATE statement will refer to the profile name which is included in the DRLTKEYS SDRLDEFS member and the two system tables (GENERATE_PROFILES and GENERATE_KEYS). For the Key Performance Metrics components, two default profiles have been created. One profile is for the IMS

Installing and uninstalling a component

KPM component (profile name of IMS, and the other profile is for the z/OS, DB2, and CICS KPM components, which collect their data from SMF (profile name of SMF).

Note that if you use the default profile name of SMF or IMS, no customizations are required in the GENERATE statements which create the tablespaces, tables, and indexes for the KPM components.

If you want to use a different profile name, you will need to customize all the GENERATE statements by copying the definitions members into your LOCAL.DEFS data set, and modifying the profile names accordingly.

If you want to use the default profile names but with a different set of table space parameters, you will need to update the GENERATE_PROFILES and GENERATE_KEYS system tables with your new table space settings for the default profiles.

Working with a component definition

About this task

This section describes these tasks:

- Controlling objects that you have modified
- Viewing objects in a component
- Viewing or editing an object definition
- Adding an object to a component
- Deleting an object from a component
- Excluding an object from a component installation
- Including an object in a component installation
- Deleting a component
- Creating a component

Controlling objects that you have modified

About this task

The variable *VERSION*, together with the *VERSION* column in the system tables, is used to:

- Ensure that unchanged Tivoli Decision Support for z/OS objects are not replaced when a component is migrated
- Provide for the control of Tivoli Decision Support for z/OS objects that you have changed

The variable *VERSION* has the value *IBM.nnn[APAR_number]*, where *nnn* is the version, release, and modification level (for example, *IBM.182* is an object supplied with Tivoli Decision Support for z/OS version 1 release 8 modification level 2). The value of *VERSION* is set for all objects when the object is installed (see “How Tivoli Decision Support for z/OS controls object replacement” on page 72 for details).

If you change a Tivoli Decision Support for z/OS-supplied object, you *must* set the variable *VERSION* to a custom version number as defined in “Tivoli Decision Support for z/OS Version variable format” on page 73. During component installation, the product can then recognize an object as having been modified by

you. When you select the component you wish to install (from the **Components** window) and press F6=Install, the **User Modified Objects** window is *automatically* displayed, listing the product-supplied objects that you have later modified.

Viewing objects in a component

About this task

You can use the administration dialog to view a list of objects in a component. To view objects in a component:

Procedure

1. From the Components window, select the component, and press Enter.
The Component window is displayed (Figure 49) for the component. All Tivoli Decision Support for z/OS objects in the component are listed..

```

                                SAMPLE Component                                ROW 1 TO 11 OF 12
Select an object. Then press Enter to Edit definition.
Description . . . . . Sample Component
Installation time . . :
Installed by . . . . . :

/ Object Name      Object Type  Member      Part
- DRLSSAMP         TABSPACE   DRLSSAMP
- SAMPLE          LOG        DRLLSAMP
- SAMPLE          REPGROUP   DRLOSAMP
- SAMPLE_H        TABLE     DRLTSAMP
- SAMPLE_H_M      UPDATE     DRLTSAMP
- SAMPLE_M        TABLE     DRLTSAMP
- SAMPLE_USER     LOOKUP     DRLTSAMP
- SAMPLE_01       RECORD     DRLRSAMP
- SAMPLE_01_H    UPDATE     DRLTSAMP
- SAMPLE01        REPORT     DRLOSAMP
- SAMPLE02        REPORT     DRLOSAMP

Command ==>
F1=Help   F2=Split   F3=Exit   F4=Exclude F5=Add obj F7=Bkwd
F8=Fwd    F9=Swap    F10=View  F11=Delete F12=Cancel
    
```

Figure 49. Component window

2. Press F10 to limit the list of objects displayed in the window.
The View Objects window is displayed.
3. Type selection criteria in fields in the View Objects window and press Enter.
Tivoli Decision Support for z/OS returns to the Component window and shows only those objects that meet the criteria.
4. You can choose to edit objects, add objects, or delete objects. When you finish, press F3.
Tivoli Decision Support for z/OS returns to the Components window.

Viewing or editing an object definition

About this task

Before you modify any data set that contains Tivoli Decision Support for z/OS definitions, copy the member to avoid changing the shipped version. Copy any member you plan to change from the product definitions or reports library to your

Working with a component definition

local definitions library, DRL.LOCAL.DEFS. (The default names of the product definitions and reports libraries are DRL182.SDRLDEFS and DRL182.SDRLRENU.)

You can use the administration dialog to view and edit an object definition. To edit an object in a component:

Procedure

1. From the Component window, select an object to work with, and press Enter. Tivoli Decision Support for z/OS accesses the ISPF editor, where you can edit (or view) the object definition.
2. When you finish editing the object definition, press F3 to exit the ISPF edit session.
Tivoli Decision Support for z/OS returns to the Component window.

Adding an object to a component

About this task

Components include object definitions necessary to collect log data, store it in the product database, and generate reports. However, if you create customized objects, you can add the object definition to an existing component.

Before using the administration dialog to add an object to a component, create the definition member that defines the object. See Chapter 4, “Overview of Tivoli Decision Support for z/OS objects,” on page 71 for more information about definition members.

To add an object to a component:

Procedure

1. From the Component window, press F5.
The Add Object window is displayed.
2. Type information about the new object, and press Enter.
You must use the same name in the Object name field as the one that appears in the definition member for the object. For example, if there is a definition member, DRLLSAMP, that contains the log collector language statement `DEFINE LOG SAMPLE;`, you must specify `SAMPLE` as the name of the log definition object.
Tivoli Decision Support for z/OS saves the object specification (that includes the name of the member that defines it) and returns to the Component window.
3. Repeat this procedure to add additional objects.

Deleting an object from a component

About this task

Components include object definitions necessary to collect log data, store it in the Tivoli Decision Support for z/OS database, and generate reports. If you do not need to collect, store, or report on certain types of data, you can delete object definitions for those data types.

Note: When you delete an object using the dialog, Tivoli Decision Support for z/OS deletes references to the object from the component. It does not delete the

definition member that contains log collector language statements that define the object. You can add the object again at a later time.

To delete an object from a component:

Procedure

1. From the Component window, select the object to delete, and press F11.
A Confirmation window is displayed.
2. From the Confirmation window, press Enter to confirm the deletion.
Tivoli Decision Support for z/OS deletes from its system tables all references from the component to the object and returns to the Component window.

Excluding an object from a component installation

About this task

This window User Modified Objects allows you to exclude product objects that have been modified by you, from the installation of the component.

Objects that are listed here were previously included by you in the component installation, although they contain your modifications to the IBM-supplied object.

For an explanation of the use of *VERSION* variable in controlling the excluding of user-modified objects from component installation, see “How Tivoli Decision Support for z/OS controls object replacement” on page 72.

To exclude an object from a component installation:>

Procedure

1. From the Components window, select the component. Then select the Show user objects option in the Component pull-down.
2. From the User Modified Objects window, select the object to exclude, and press F4.
A Confirmation window is displayed.
3. From the Confirmation window, press Enter to confirm that the object should be excluded from the installation.

Including an object in a component installation

About this task

After you have excluded an object from the installation of a component (see “Excluding an object from a component installation” for details), you have the option to re-include the object.

To include an object in a component installation:

Procedure

1. From the Components window, select the component. Then select the Show excluded option in the Component pull-down.
2. From the Objects Excluded window, select the object to include, and press F4.
A Confirmation window is displayed.
3. From the Confirmation window, press Enter to confirm that the object should be included in the installation.

Deleting a component

About this task

To remove all references to a component from Tivoli Decision Support for z/OS, you can use the administration dialog to delete the component. Do not delete components shipped with the product unless you are sure you are not going to use them.

To delete a component:

Procedure

1. Uninstall the component that you plan to delete. See “Uninstalling a component” on page 134 for more information.
You must uninstall a component before deleting it. Uninstalling deletes all objects of the component.
2. From the Components window, select the component. Then select the Delete option in the Component pull-down.

A confirmation window is displayed.

3. Press Enter to confirm the deletion.

Tivoli Decision Support for z/OS deletes from its system tables all references to the component. The component no longer appears in the list of components in the Components window. The feature definition member (see Chapter 4, “Overview of Tivoli Decision Support for z/OS objects,” on page 71) still exists, however, and you can reinstall it at a later time. Before reinstalling deleted components, you must update the system tables to refresh the list of components available for installation.

Creating a component

About this task

If you have created a set of definitions (for example, for records or tables) using log collector language or report definition language, you can package them as a component. Creating a component can also be useful when designing a component to use at other sites. You must also transfer members that define the objects to the system at the other site.

You can define a component with SQL statements that directly update these system tables: DRLCOMPONENTS, DRLCOMP_PARTS, and DRLCOMP_OBJECTS, described in “Dialog system tables” on page 235. Tivoli Decision Support for z/OS features define entries in these tables as you create or update the system tables, using SQL statements in definition members. For examples of component definition members, see Chapter 4, “Overview of Tivoli Decision Support for z/OS objects,” on page 71.

Note: As you create your component, remember that the product requires that some definitions exist before you can install others. For example, if your component contains record procedures, you must install the record definition that maps the source record for the record procedure before installing the record procedure. Furthermore, you must install the record procedure before installing the record definition that maps the output of the record procedure. To do this, put both definitions in the same member.

Tivoli Decision Support for z/OS installs component definitions in the following order:

1. Log
2. Record
3. Record procedure
4. Tablespace
5. Lookup table
6. Table
7. Update
8. View
9. Report group
10. Report

The order of installation within a definition type is determined by the sorting sequence of the definition member names.

If you plan to use a component on the same Tivoli Decision Support for z/OS system on which you are creating it, you can use the administration dialog to create the component.

Procedure

1. Optionally, you can select an existing component for Tivoli Decision Support for z/OS to use as a template for the new component before performing the next step.
2. From the Components window, press F5.
The New Component window is displayed.
3. Type information about the new component in the fields.
4. Press F5 to add an object to the component.
The Add Object window is displayed. See “Adding an object to a component” on page 138 for more information.
5. Select an object, and press Enter to edit its definition.
Tivoli Decision Support for z/OS accesses the ISPF editor, where you can edit the object definition. See “Viewing or editing an object definition” on page 137 for more information.
6. To delete an object that currently exists (either it existed in the template or you decided not to use an object you added), select the object, and press F11.
A Confirmation window is displayed for you to confirm the deletion. See “Deleting an object from a component” on page 138 for more information.
7. When you finish adding, editing, or deleting objects, press F3.
Tivoli Decision Support for z/OS returns to the Components window and lists the new component.

Working with a component definition

Chapter 8. Working with log and record definitions

Tivoli Decision Support for z/OS uses log definitions to associate a series of processing definitions with a certain type of log data set. An example is the SMF log definition that the product uses to process SMF log data sets created by MVS. Tivoli Decision Support for z/OS associates log, record, and update definitions with the SMF log and uses these definitions to collect the data, manipulate it, and store it in appropriate tables.

This section describes how to use the administration dialog to work with log and record definitions. It describes how to:

- Work with the contents of logs (page “Working with the contents of logs” on page 144):
 - View a list of log data sets that Tivoli Decision Support for z/OS has collected (page “Viewing a list of log data sets collected” on page 144)
 - Collect data from a log into DB2 tables (page “Collecting data from a log into DB2 tables” on page 146)
 - Display statistics of log data sets (page “Displaying log statistics” on page 148)
 - Display the contents of a log data set (page “Displaying the contents of a log” on page 149)
 - Generate a report on a record in a log data set (page “Creating a report on a record” on page 150)
- Work with log definitions (page “Working with log definitions” on page 151):
 - View and modify a log definition and its header fields (page “Viewing and modifying a log definition” on page 152)
 - Create a log definition (page “Creating a log definition” on page 153)
 - Delete a log definition (page “Deleting a log definition” on page 154)
- Work with record definitions (page “Working with record definitions in a log” on page 154):
 - View and modify a record definition (page “Viewing and modifying a record definition” on page 155):
 - Work with fields in a record definition (page “Working with fields in a record definition” on page 157)
 - Work with sections in a record definition (page “Working with sections in a record definition” on page 158)
 - Create a record definition (page “Creating a record definition” on page 159)
 - Display update definitions associated with a record (page “Displaying update definitions associated with a record” on page 159)
 - Delete a record definition (page “Deleting a record definition” on page 160)
 - View and modify a record procedure definition (page “Viewing and modifying a record procedure definition” on page 160)
 - Create a record procedure definition (page “Creating a record procedure definition” on page 161)
 - Delete a record procedure definition (page “Deleting a record procedure definition” on page 162)

Working with the contents of logs

About this task

To work with logs, first display a list of log definitions stored in the product system tables.

Procedure

1. From the Tivoli Decision Support for z/OS Administration window, select 3, Logs.
2. Press Enter.
Tivoli Decision Support for z/OS displays the Logs window.

Viewing a list of log data sets collected

About this task

The product Data Sets window shows you a list of data sets that have been collected. The window (Figure 50 on page 145) shows the name of each data set, when it was collected, and the status of the collect job.

The Status column reads **OK** if the collect job ran uninterrupted and without error. It shows **Incomplete** if the job was interrupted before the entire log had been processed. For example, due to a locking or out of space problem. **Warning** in the Status column means that the collect issued warning messages but the job completed successfully.

You can display detailed collection statistics for each collected data set. This is the default action for the window; you perform it by pressing Enter after selecting a data set.

You can also display the data in a log data set, record by record.

To view a list of collected log data sets:

Procedure

1. From the Logs window, select a log definition and press F6.
Tivoli Decision Support for z/OS displays the Data Sets window for the log type you selected (see Figure 50 on page 145). You can then display collect statistics for each data set.


```

SMF Data Sets                                ROW 1 TO 15 OF 169

Select one data set.  Then press Enter to view statistics.

/  Data Sets                                Time collected      Status
-  SYST.SMFSYSA.D930131                    2000-02-01-04.26.57 OK
-  SYST.SMFSYSA.D930130                    2000-01-31-05.22.15 OK
-  SYST.SMFSYSB.D930129                    2000-01-30-04.14.36 OK
-  SYST.SMFSYSA.D930129                    2000-01-30-02.22.14 Incomplete
-  SYST.SMFSYSB.D930128                    2000-01-29-02.59.20 OK
-  SYST.SMFSYSA.D930128                    2000-01-29-01.38.50 OK
7  SYST.SMFSYSB.D930127                    2000-01-28-08.30.02 Warning
-  SYST.SMFSYSA.D930127                    2000-01-28-03.56.24 Warning
-  SYST.SMFSYSB.D930126                    2000-01-27-03.23.27 OK
-  SYST.SMFSYSA.D930126                    2000-01-27-03.26.17 OK
-  IVT.SMFCICS.TEST1                      2000-01-26-14.23.23 OK
-  IVT.SMFCICS.DELTA                      2000-01-26-10.42.26 OK
-  SYST.SMFSYSB.D930125                    2000-01-26-04.18.48 OK
-  SYST.SMFSYSA.D930125                    2000-01-26-02.56.26 OK
-  SYST.SMFSYSB.D930124                    2000-01-26-04.18.48 OK

Command ==>
F1=Help      F2=Split    F3=Exit      F5=Display   F7=Bkwd      F8=Fwd
F9=Swap      F11=Delete  F12=Cancel

```

Figure 50. Data Sets window

- From the Data Sets window, select a data set and press Enter.
The product displays the Collect Statistics window for the data set (Figure 51).

```

SMF Collect Statistics

Press Enter to return.

Data set . . . . : SYST.SMFSYSB.D930127
Volume . . . . . : TS0007

Time collected . : 2000-01-28-08.30.02  Collected by . . . : STROMBK
Elapsed time . . : 462                  Return code . . . : 4
Times collected . : 1                   Completed . . . . . : Y

First record . . : 00001E2900006EB60093104FD3C4C7F140404040
                  : 0003000500000000004400180001000000000000
First timestamp . : 2000-01-27-00.04.43
Last timestamp . : 2000-01-27-22.17.23

Records read . . : 187714                Records selected . : 17701

Database updates : 0                    Inserts : 13610    Deletes : 0

F1=Help      F2=Split    F9=Swap      F12=Cancel

```

Figure 51. Collect Statistics window

- Press Enter to return to the Data Sets window after you finish viewing statistics.

What to do next

To display the contents of a data set record by record, select the data set and press F5.

Working with the contents of logs

Tivoli Decision Support for z/OS displays the Record Selection window. Refer to “Displaying the contents of a log” on page 149 for more information.

Deleting a log data set

About this task

To delete data set statistics from the product system tables:

Procedure

1. From the Data Sets window, select the data set and press F11.
Tivoli Decision Support for z/OS displays a confirmation window.
2. Press Enter to confirm the deletion.
The product deletes any references it has to the data set, which no longer appears in the list of collected data sets.

Collecting data from a log into DB2 tables

About this task

Tivoli Decision Support for z/OS stores data it collects in DB2 tables in the product database, following the instructions in update definitions associated with records in the log.

Usually, you use a batch job to collect log data. (See “Collecting log data” on page 85 for more information about sample collect jobs.) However, you can use the administration dialog to perform online collection. For example, to correct problems or to test new log, record, or update definitions.

Note: Some logs require special processing or contain collect statements that can be initiated only from batch jobs. Such logs include those for DCOLLECT, VMACCT, SMF_VPD, and IMS.

To collect data from a log into DB2 tables:

Procedure

1. From the Logs window, select a log and press F11.
The Collect window is displayed (see Figure 52 on page 147).

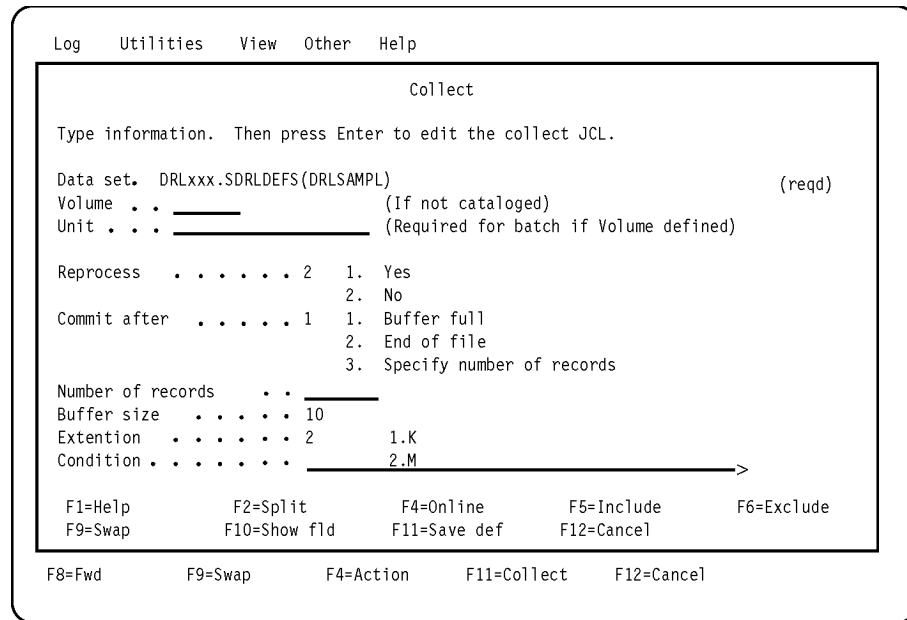


Figure 52. Collect window

2. Type the name of the log data set in the Data set field.

Note: The log data sets used as input for the collect (DRLLOG DD statement) are expected to be sorted in chronological order).

3. Optionally, specify other collect options in fields in the window.

Note: Entry fields followed by a greater than (>) sign respond to the F10 (Show fld) function key, which displays all of the data in the field or lets you type more data in the Show Field window.

4. Press F5 to include only specific DB2 tables in the collect process.
The Include Tables window is displayed.
5. Select those tables to include in the collect process and press Enter.
You are returned to the Collect window.

You can exclude tables as well. You need exclude only tables that the product would normally update during the collection.

6. Press F6 to exclude tables from the collect process.
The Exclude Tables window is displayed. Select tables to exclude from the collect process and press Enter. You are returned to the Collect window.

7. Run the collect either in batch or online:

- a. Press Enter to run the collect in batch mode.

Tivoli Decision Support for z/OS builds a JCL job stream for the collect job and accesses the ISPF editor where you can edit and submit the JCL.

- b. Press F4 to perform an online collection.

Tivoli Decision Support for z/OS starts the collect process online. When the collection is complete, collect messages are displayed in an ISPF browse window.

8. Press F3 to return to the Logs window.

Displaying log statistics

About this task

You can create log statistics for any log data set, regardless of whether it has been collected. A log statistics file shows the number of records of each type in a log data set. It also shows records built by log and record procedures.

To view statistics for a log data set:

Procedure

1. From the Logs window, select a log definition.
2. Select 3, Show log statistics, from the Log pull-down.
You are prompted for the name of a log data set.
3. Type the name of the data set and press Enter.
The product displays statistics for the log (see Figure 11 on page 37).

```

DRLnnnnI Logstat started at 2000-12-04-10.04.15
DRL0302I Processing SMF.DATA.SET on VOL001
DRL0341I The first record timestamp is 2000-06-03-07.00.01.730000.
DRL0342I The last record timestamp is 2000-06-03-11.52.40.220000.
DRL0003I
DRL0315I Records read from the log or built by log procedure:
DRL0317I Record name          Number
DRL0318I -----
DRL0319I SMF_000                0
DRL0319I SMF_006                6
DRL0319I SMF_007                0
DRL0319I SMF_021                0
DRL0319I SMF_025                0
DRL0319I SMF_026                476
DRL0319I SMF_030               3737
DRL0319I SMF_070                40
DRL0319I SMF_071                40
DRL0319I SMF_072_1             280
DRL0319I SMF_090                0
DRL0320I Unrecognized          3
DRL0318I -----
DRL0321I Total                  4582
DRL0003I
DRL0316I Records built by record procedures:
DRL0317I Record name          Number
DRL0318I -----
DRL0319I SMF_030_X                2012
DRL0319I SMF_070_X                200
DRL0318I -----
DRL0321I Total                  2212
DRLnnnnI Logstat ended at 2000-12-04-10.09.43
    
```

Figure 53. Sample log statistics output

4. When you finish viewing statistics, press F3.
The Logs window is displayed.

Displaying the contents of a log

About this task

Tivoli Decision Support for z/OS provides a facility for displaying the contents of a log, record by record. The Record Data window describes each field in each record in the log data set you identify.

To view the contents of a log:

Procedure

1. From the Logs window, select the log.
2. From the Utilities pull-down, select 2, Display log, and press Enter.

Note: You can also display the contents of a log by selecting Display record from the Record Definition window or by pressing F5 from the Data Sets window.

The Record Selection window is displayed.

3. Type the log data set name and, optionally, the name of a record type (to display only one record definition), or a record sequence number (to start displaying records at that position in the log). Press Enter.

The Record Data window is displayed.

Record Data					ROW 1 TO 13 OF 222
Press Enter to view the next record.					
Record name . . :	SMF_030	Record number :	3		
Data set . . . :	LDG.SMFSYSA.W20				
Field Name	Type	Length	Offset	Value	
SMF30LEN	BINARY	2	0	628	
SMF30SEG	BINARY	2	2	0	
SMF30FLG	BIT	1	4	11001110	
SMF30RTY	BINARY	1	5	30	
SMF30TME	TIME	4	6	07.00.03.830000	
SMF30DTE	DATE	4	10	2000-06-03	
SMF30SID	CHAR	4	14	MVS1	
SMF30WID	CHAR	4	18	JES2	
SMF30STP	BINARY	2	22	3	
-----	-----	-----	-----	-----	
TRIPLETS	SECTION	88	24	1 (1)	
-----	-----	-----	-----	-----	
SMF30SOF	BINARY	4	0	112	
Command ==> _____					
F1=Help F2=Split F7=Bkwd F8=Fwd F9=Swap F12=Cancel					

Figure 54. Record Data window

4. Press Enter to step through records in the log.
Each time you press Enter, Tivoli Decision Support for z/OS displays the next identified record in the log.
5. When you finish viewing record data, press F12.
You are returned to the Logs window.

Creating a report on a record

About this task

To produce a report of the data in a record type without performing a collect operation, you can use the Tivoli Decision Support for z/OS list function. For example, you may need very detailed data from a record, or you may want to get information from a record one time, without creating product tables for it. The list function creates a report of the data in a record either in QMF format or as a data set that can be browsed.

To create a report of the data in a record:

Procedure

1. From the Logs window, select the log and press Enter.
The Record Definitions window for the log is displayed (see Figure 58 on page 155).
2. Select a record and press F11.
The List Record window for the record is displayed (see Figure 55).

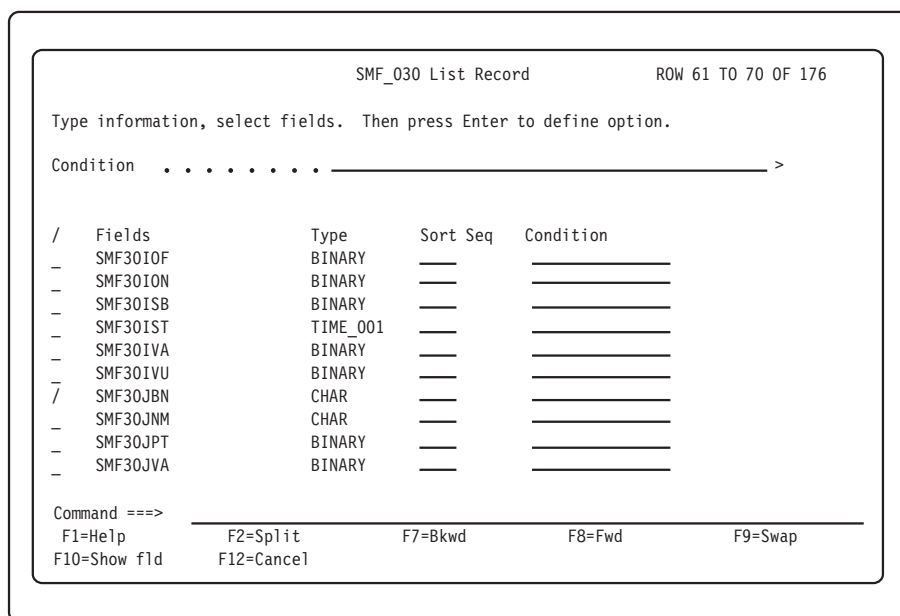


Figure 55. List Record window

3. From the List Record window, select fields to include in the report. Type information in the fields and press Enter.
If your installation uses QMF, the Report Display Options window is displayed.
4. In the Report Display Options window, choose whether to display the report using QMF or as a data set that can be browsed. Specify the name of the log data set from which Tivoli Decision Support for z/OS is to produce the report, then press Enter.
If your installation does not use QMF, the report is displayed using ISPF browse. Specify the name of the log data set from which Tivoli Decision Support for z/OS is to produce the report in the Input Log Data Set Name window, then press Enter.

The report is displayed.

```

ISRBROBF STROMBK.DRLLST1 ----- LINE 00000000 COL 001 080
***** TOP OF DATA *****
SMF30CPS  SMF30CPT  SMF30DTE  SMF30JBN  SMF30RST  SMF30SIT
-----
      0      18  2000-06-03  LOGREFL1  07.00.00  07.00.01
      1      19  2000-06-03  LOGREFL2  07.00.00  07.00.05
      0      17  2000-06-03  LOGREES1  07.00.01  07.00.07
      0      13  2000-06-03  LOGREES2  07.00.01  07.00.09
      2      20  2000-06-03  LOGRSP4A  07.00.02  07.00.10
      1      19  2000-06-03  LOGSP4B  07.00.02  07.00.22
      0      16  2000-06-03  LOGRXAA  07.00.03  07.00.23
      0      13  2000-06-03  LOGRXAB  07.00.03  07.00.26
      4      73  2000-06-03  EID3D105  07.00.12  07.00.13
      0       7  2000-06-03  EID3D105  07.00.12  07.01.21
      9      79  2000-06-03  EID3D105  07.00.12  07.01.21
    227    1108  2000-06-03  EID4      01.14.42  01.14.43
      18     226  2000-06-03  EID4      07.12.42  07.12.44
      1      12  2000-06-03  XGORANW  07.26.33  07.26.34
      1      12  2000-06-03  XGORANW  07.26.50  07.26.51
      7      215  2000-06-03  NORBACK  07.31.52  07.31.52

The list record action is executed successfully.
COMMAND ==>>
F1=Help      F2=          F3=End      F4=          F5=R Find   F6=R Change
F7=Backward F8=Forward  F9=         F10=Left    F11=Right  F12=Cursor
    
```

Figure 56. Output from List record function

5. When you finish viewing the report, press F3 to exit QMF or the ISPF browse window.
You are returned to the List Record window.
6. From the List Record window, press F12 to return to the Record Definitions window.
7. From the Record Definitions window, repeat this procedure for more records or press F3 to return to the Logs window.

Working with log definitions

All the logs that you plan to process must be defined to Tivoli Decision Support for z/OS. Log definitions included with each component define the logs that the product uses to collect data.

A log definition can include these elements:

Header

Lists fields common to all records in the log.

Timestamp

Describes how to derive the timestamp of a record from fields in the header.

First record

Describes a condition that should be met for the first record in the log data set.

Last record

Describes a condition that should be met for the last record in the log data set.

Log procedure

Identifies a program that is invoked for each record read.

Working with log definitions

Log procedure parameters

Identifies the language of the log procedure and other information, such as information the log procedure cannot retrieve from the record.

For more information about log procedures, refer to the *Language Guide and Reference*.

Viewing and modifying a log definition

About this task

You can use the administration dialog to view or modify log definitions. To view and modify a log definition:

Procedure

1. From the Logs window, select the log and press F5.
The Log Definition window is displayed (see Figure 57) for the log you specified.

```
Log Utilities View Other Help
SMF Log Definition
Type information. Press Enter to save and return.
Description . . . . . MVS Systems Management Facility >
Log procedure . . . . . _____ >
Log procedure parameter . . . _____ >
Log procedure language . . . 1. ASM
                                   2. C
Record timestamp . . . . . TIMESTAMP(SMFDTE, SMFTME) >
First record condition . . . SMFRTY = 2 >
Last record condition . . . SMFRTY = 3 >
F1=Help      F2=Split      F5=Header      F9=Swap      F10=Show fld
F12=Cancel

Command ==>
F1=Help      F2=Split      F3=Exit      F5=Log def   F6=Datasets  F7=Bkwd
F8=Fwd      F9=Swap      F10=Actions  F11=Collect  F12=Cancel
```

Figure 57. Log Definition window

2. Change the log definition.
3. Press F5 to display header fields for the log definition.
The Header Fields window is displayed for the log. See “Working with header fields” for more information.
4. When you finish modifying the log definition, press Enter.
The changes are saved and you are returned to the Logs window.

Working with header fields

About this task

To add header fields to a log definition:

Procedure

1. From the Header Fields window, press F5 to add a header field.
A blank Header Field Definition window is displayed.
2. Type the required information in the fields and press Enter.
The Header Field Definition window for the next field is displayed. Tivoli Decision Support for z/OS carries forward values for the Type and Length fields from the previous field and increments the Offset field by the length of the previous field.
3. Press F12 when you finish adding fields.
You are returned to the Header Fields window.
4. Press F3 to return to the Log Definition window.

Modifying header fields for a log definition**About this task**

To modify header fields for a log definition:

Procedure

1. From the Header Fields window, select the header field and press Enter.
The Header Field Definition window for the header field you specified is displayed.
2. Type changes in the fields and press Enter.
You are returned to the Header Fields window.
3. Press F3 to return to the Log Definition window.

Deleting header fields for a log definition**About this task**

To delete header fields for a log definition:

Procedure

1. To delete a header field, select the field and press F11.
A confirmation window is displayed.
2. Press Enter to confirm the deletion.
The header field is deleted from the list and you are returned to the Header Fields window.
3. Press F3 to return to the Log Definition window.

Creating a log definition**About this task**

To collect data from a log that is not defined by a Tivoli Decision Support for z/OS component, you must create a log definition. You can use the administration dialog to create log definitions, or you can use the log collector language. Refer to the *Language Guide and Reference* for more information about creating log definitions with log collector language.

To create a log definition:

Working with log definitions

Procedure

1. To use an existing log definition as a template, select a log definition from the Logs window. Otherwise, do not select a log definition before the next step.
2. Select 1, New, from the Log pull-down and press Enter.
The New Log Definition window is displayed.
3. Type information for the new log definition in the fields.
4. Press F5 to add header fields to the log definition.
The Header Fields window is displayed. See “Working with header fields” on page 152 for more information on adding header fields.
5. After you add all the information, press Enter.
The new log definition is saved and you are returned to the Logs window.

Deleting a log definition

About this task

If you no longer need to collect data from a log, you can use the administration dialog to delete the log definition. When you delete this log definition, you delete references to the log definition from Tivoli Decision Support for z/OS system tables, but you do not delete the member that defines the log type.

To delete a log definition:

Procedure

1. From the Logs window, select a log and then select the Delete option from the Log pull-down.
A confirmation window is displayed.
2. Press Enter to confirm the deletion.
The log definition is deleted and you are returned to the Logs window.

Working with record definitions in a log

About this task

Each record in a log belongs to a record type that must be defined to Tivoli Decision Support for z/OS to be collected. Otherwise, the product designates it as an unrecognized type of record and does not process it. Record definitions are included with each predefined component.

To view a list of record definitions:

Procedure

1. From the product Administration window, select 3, Logs, and press Enter.
The Logs window is displayed.
2. From the Logs window, select the log that contains the record and press Enter.
The Record Definitions window for the log is displayed (see Figure 58 on page 155).

```

Record  Utilities  Other  Help
-----
                          SMF Record Definitions                          ROW 8 TO 20 OF 124

Select a record definition.  Then press Enter to Open record definition.

/  Record Definitions  Description
-  SMF_000              IPL
-  SMF_002              Dump header
-  SMF_003              Dump trailer
-  SMF_004              Step termination
-  SMF_005              Job termination
-  SMF_006              JES2/JES3/PSF/External writer
-  SMF_007              Data lost
-  SMF_008              I/O configuration
-  SMF_009              VARY device ONLINE
-  SMF_010              Allocation recovery
-  SMF_011              VARY device OFFLINE
-  SMF_014              INPUT or RDBACK data set activity
-  SMF_015              OUTPUT, UPDAT, INOUT, or OUTIN data set

Command ==>
F1=Help   F2=Split   F3=Exit   F5=Procs   F6=Updates  F7=Bkwd
F8=Fwd    F9=Swap    F10=Actions F11=List rec F12=Cancel

```

Figure 58. Record Definitions window

Viewing and modifying a record definition

About this task

Most of a record definition describes the layout of the record. Records are divided into fields and, optionally, sections. A field is a named sequence of adjacent bytes. A section is a larger structure that contains fields or other sections. For more information about defining records, sections, and fields, refer to the *Language Guide and Reference*.

You can use the administration dialog to modify record definitions. To view and modify a record definition:

Procedure

1. From the **Record Definitions** window, select the record definition and press Enter.

The **Record Definition** window for the record definition is displayed (see Figure 59 on page 156).

Working with record definitions in a log

```

SMF_030 Record Definition                                ROW 1 TO 9 OF 188

Type information. Select a field or a section. Then press Enter to
display.

Log name . . . SMF
Identified by . SMF30RTY= 30                            > (condition)
Built by . . . _____ + (program name)
Description . . Common address space work                >

/  Field                Type      Length  Offset  Section
-  SMF30LEN             BINARY   2       0
-  SMF30SEG             BINARY   2       2
-  SMF30FLG             BIT       1       4
-  SMF30RTY             BINARY   1       5
-  SMF30TME             TIME_001  4       6
-  SMF30DTE             DATE_001  4      10
-  SMF30SID             CHAR     4      14
-  SMF30WID             CHAR     4      18
-  SMF30STP             BINARY   2      22

Command ==> _____
F1=Help   F2=Split   F3=Exit   F4=Prompt   F5=Add fld   F6=Add sec
F7=Bkwd   F8=Fwd     F9=Swap   F10=Show fld F11=Delete  F12=Cancel

```

Figure 59. Record Definition window

2. Type any changes to the record definition.

Note: By changing the value in the Log name field, you can move the record to another log definition.

3. To modify the definition of a field, select the field and press Enter.
The **Field Definition** window is displayed. See “Working with fields in a record definition” on page 157 for more information.
4. To modify a section, select the section and press Enter.
The **Section Definition** window is displayed. See “Working with sections in a record definition” on page 158 for more information.
5. Press F5 to add fields to the record definition.
The **Field Definition** window is displayed. See “Working with fields in a record definition” on page 157 for more information.
6. Press F6 to add sections to the record definition.
The **Section Definition** window is displayed. See “Working with sections in a record definition” on page 158 for more information.
7. To delete a section or field from the record definition, select the section or field and press F11.
If the section or field definition already existed in the record definition, a confirmation window is displayed. Otherwise, you are deleting something you just added. Tivoli Decision Support for z/OS does not ask you to confirm this type of deletion and you can skip step 8.
8. Press Enter to confirm the deletion.
The section or field is deleted and you are returned to the Record Definition window.
9. Press F3 when you finish modifying the record definition.
Your changes are saved and you are returned to the Record Definitions window.

Note: If you have incorrectly modified the record definition, Tivoli Decision Support for z/OS displays error messages in an ISPF browse window. Examine the messages and press F3 to return to the Record Definition window where you can correct the errors.

Working with fields in a record definition

About this task

You can use the administration dialog to modify existing field definitions or to add field definitions. You can also use log collector language statements. Refer to the *Language Guide and Reference* for more information about defining fields in a record.

To add a field definition to a record definition:

Procedure

1. From the Record Definition window, press F5.
A blank Field Definition window is displayed.
2. Type the required information in the fields and press Enter.
Another Field Definition window is displayed (see Figure 60).

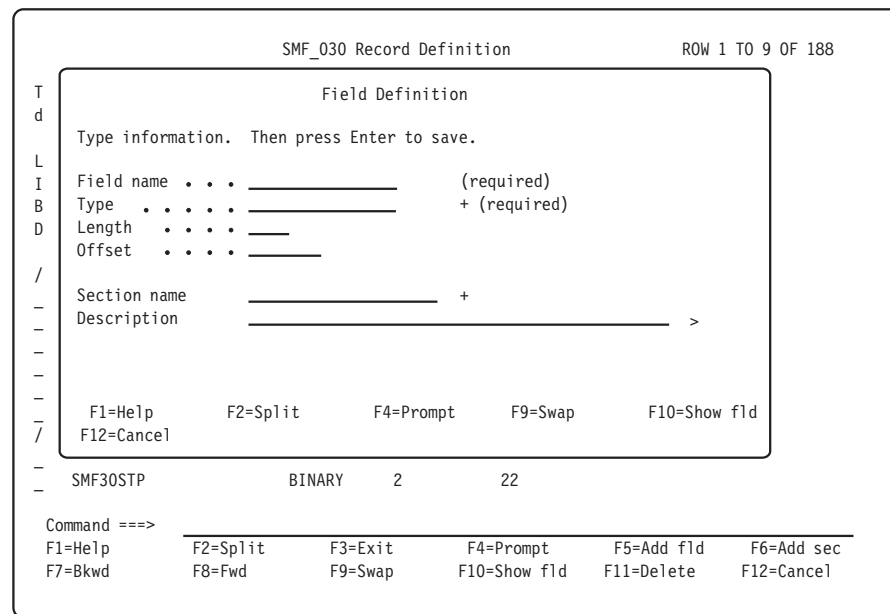


Figure 60. Field Definition window

3. Press F12 when you finish adding fields.
You are returned to the Record Definition window.

Modifying a field definition

About this task

To modify a field definition:

Procedure

1. From the Record Definition window, select the field and press Enter.
The Field Definition window is displayed.

Working with record definitions in a log

2. Type changes in the fields and press Enter.
Your changes are saved and you are returned to the Record Definition window.

Working with sections in a record definition

About this task

You can use the administration dialog to modify existing section definitions or to add section definitions. You can also use log collector language statements. Refer to the *Language Guide and Reference* for more information about defining sections and repeated sections.

To modify a section definition:

Procedure

1. From the Record Definition window, select the section and press Enter.
The Section Definition window is displayed (see Figure 61).

```
SMF_030 Record Definition                                ROW 42 TO 50 OF 188

Section Definition

Type information. Press Enter to save and return.

Section name . . . . SUBSYSTEM                          (required)
In Section name . . . _____ +

Offset . . . . . SMF30SDF                               >(required)
Length . . . . . SMF30SLN                               >
Number . . . . . SMF30SON                               >
Present if condition _____                       >

Repeated . . . . . 2  1. Yes
                   2. No

F1=Help      F2=Split      F4=Prompt      F9=Swap      F10=Show fld
F12=Cancel

Command ==>
F1=Help      F2=Split      F3=Exit      F4=Prompt      F5=Add fld      F6=Add sec
F7=Bkwd      F8=Fwd       F9=Swap      F10=Show fld  F11=Delete     F12=Cancel
```

Figure 61. Section Definition window

2. Type changes in the fields and press Enter.
Your changes are saved and you are returned to the Record Definition window.

Adding a section definition to a record definition

About this task

To add a section definition to a record definition:

Procedure

1. From the Record Definition window, press F5.
A blank Section Definition window is displayed.
2. Type the required information in the fields and press Enter.
Another Section Definition window is displayed.
3. Press F12 when you finish adding sections.

You are returned to the Record Definition window.

Creating a record definition

About this task

You can create record definitions by using:

- The administration dialog; or
- Log collector language statements.

For more information about defining records with the log collector language, refer to the *Language Guide and Reference*.

To create a record definition:

Procedure

1. To use an existing record definition as a template, select a record definition from the Record Definitions window. Otherwise, do not select a record definition.
2. From the Record Definitions window, select 1, New, from the Record pull-down.
The New Record Definition window is displayed.
3. Type information for the new record definition in fields of the window.
4. Press F5 to add fields to the record definition.
The Field Definition window is displayed. See “Working with fields in a record definition” on page 157 for more information.
5. Press F6 to add sections to the record definition.
The Section Definition window is displayed. See “Working with sections in a record definition” on page 158 for more information.
6. Press F3 when you finish adding fields and sections.
The new record definition is saved and you are returned to the Record Definitions window.

Displaying update definitions associated with a record

About this task

Update definitions contain instructions for summarizing log data into DB2 tables. The Record Definitions window lets you view which update definitions Tivoli Decision Support for z/OS uses to process data that a record definition maps.

Each record is associated with one or more update definitions. To display update definitions associated with a record:

Procedure

1. From the Record Definitions window, select the record with associated update definitions you plan to view and press F6.
The Update Definitions window lists all the update definitions that use the selected record definition as input. From this window, you can view, modify, or add update definitions. See “Displaying and modifying update definitions of a table” on page 184 or “Creating an update definition” on page 201 for more information.
2. Press F3 when you finish viewing update definitions.
You are returned to the Record Definitions window.

Deleting a record definition

About this task

If you no longer require data from a certain record, you can use the administration dialog to delete the record definition.

Note: Tivoli Decision Support for z/OS prevents you from deleting record definitions that affect, or are affected by, other Tivoli Decision Support for z/OS objects. To delete a record definition, remove links from it to other Tivoli Decision Support for z/OS objects.

To delete a record definition:

Procedure

1. From the Record Definitions window, select the record definition to delete. Then select 5, Delete, from the Record pull-down.
A confirmation window is displayed.
2. Press Enter to confirm the deletion.
The record definition is deleted and you are returned to the Record Definitions window.

Viewing and modifying a record procedure definition

About this task

Record procedures are programs that can modify, split, combine, sort, delete, or perform any function to records during collection. Record procedures use existing records as input and produce other records, which must be defined to Tivoli Decision Support for z/OS. Some product components include record procedures and their definitions.

Each record procedure definition defines record types that the procedure processes, identifies the language of the procedure, and passes parameters to the procedure. For more information, refer to the *Language Guide and Reference*.

You can use the administration dialog to modify record procedure definitions.

To view and modify a record procedure definition:

Procedure

1. From the Record Definitions window, select the record definition that is input to the record procedure you plan to modify and press F5.
The Record Procedures window for the record definition is displayed. This window lists all record procedure names that use the record as input.
2. From the Record Procedures window, select the record procedure whose definition you plan to modify and press Enter.
The Record Procedure Definition window for the record procedure is displayed (see Figure 62 on page 161).

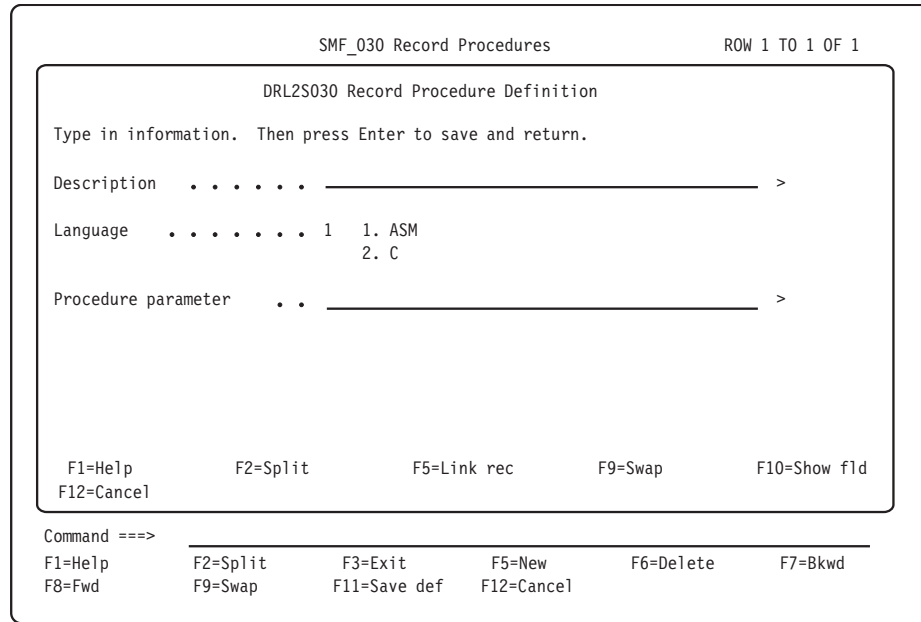


Figure 62. Record Procedure Definition window

3. Type your changes in the fields.
4. Press F5 to link record definitions to the record procedure (to define them as input to the record procedure).
The Record Definitions window is displayed.
5. From the Record Definitions window, select record definitions to link to the record procedure and press Enter.
The record procedure is linked to the record definitions you selected and you are returned to the Record Procedure Definition window.
6. When you finish modifying the record procedure definition, press Enter.
Your changes are saved and you are returned to the Record Procedures window.
7. Repeat this procedure for other record procedures or press F3 to return to the Record Definitions window.

Creating a record procedure definition

About this task

If you must add a record procedure, you must first write a program according to the instructions in the *Language Guide and Reference*. You can then use the administration dialog to define the record procedure to Tivoli Decision Support for z/OS.

To create a record procedure definition:

Procedure

1. From the Record Definitions window, select the record definition from which the new record procedure derives its input and press F5.
The Record Procedures window for the record definition is displayed.
2. From the Record Procedures window, press F5.
The New Record Procedure Definition window is displayed.

Working with record definitions in a log

3. Type information for the new record procedure in the fields.
4. Press F5 if you want to link the record procedure to additional record definitions that describe record types on which the record procedure acts. The record procedure is automatically linked to the record type selected in step 1 above.

The Record Definitions window is displayed.

5. From the Record Definitions window, select record definitions to link to the record procedure and press Enter.

The record procedure is linked to the record definitions you selected and you are returned to the Record Procedure Definition window.

6. When you finish entering information, press Enter.

The new record procedure is saved and returns to the Record Procedures window.

7. Repeat this procedure to add more record procedures or press F3 to return to the Record Definitions window.

What to do next

In addition, you must define a record type as the record procedure's output. Do this in the Record Definition window (Figure 59 on page 156). Type the record procedure name in the Built by field, to identify a record type as one that is created by the record procedure.

Deleting a record procedure definition

About this task

If you no longer require a record procedure, you can use the administration dialog to delete the record procedure definition.

Note: Tivoli Decision Support for z/OS prevents you from deleting record procedure definitions that affect, or are affected by, other product objects. To delete a record procedure definition, remove links from the record procedure to other product objects.

To delete a record procedure definition:

Procedure

1. From the Record Definitions window, select the record definition that is associated with the record procedure to delete and press F5.
The Record Procedures window for the record definition is displayed.
2. From the Record Procedures window, select the record procedure to delete and press F6.
A confirmation window is displayed.
3. Press Enter to confirm the deletion.
You are returned to the Record Procedures window.
4. Repeat this procedure to delete more record procedures or press F3 to return to the Record Definitions window.

Results

The record procedure is deleted.

Chapter 9. Working with tables and update definitions

This section describes how to use the administration dialog to work with tables, update definitions, and other table-related objects such as purge conditions, indexes, views, and table spaces. After reading this section, you should be familiar with these tasks:

- “Working with data in tables” on page 164
 - “Displaying the contents of a table” on page 164
 - “Editing the contents of a table” on page 165
 - “Showing the size of a table” on page 167
 - “Recalculating the contents of a table” on page 169
 - “Importing the contents of an IXF file to a table” on page 171. (This option is available only if your installation uses QMF with Tivoli Decision Support for z/OS.)
 - “Exporting table data to an IXF file” on page 172. (This option is available only if your installation uses QMF with Tivoli Decision Support for z/OS.)
 - “Purging a table” on page 172
 - “Unloading and loading tables” on page 172
- “Working with tables and update definitions” on page 179
 - “Opening a table to display columns” on page 179
 - “Displaying and modifying update definitions of a table” on page 184
 - “Displaying and editing the purge condition of a table” on page 190
 - “Displaying and modifying a table or index space” on page 192
 - “Displaying a view definition” on page 196
 - “Printing a list of Tivoli Decision Support for z/OS tables” on page 197
 - “Saving a table definition in a data set” on page 197
 - “Listing a subset of tables in the Tables window” on page 198
 - “Creating a table” on page 198
 - “Deleting a table or view” on page 200
 - “Creating a table space” on page 200
 - “Creating an update definition” on page 201
 - “Deleting an update definition” on page 202
 - “Administering user access to tables” on page 202

When you use Tivoli Decision Support for z/OS to collect log data, the product stores the data in DB2 tables in its database. To view a list of the tables that are used to store collected data, from the **Administration** window, select 4, Tables. The **Tables** window is displayed. The list in this window includes all the product data tables, lookup tables, and control tables.

Working with tables and update definitions

```
Table Maintenance Utilities Edit View Other Help
-----
                Tables                ROW 1 TO 13 OF 212

Select one or more tables.  Then press Enter to Open table definition.

/  Tables      Prefix  Type
-  AVAILABILITY_D  DRL    TABLE
-  AVAILABILITY_M  DRL    TABLE
-  AVAILABILITY_PARM DRL    TABLE
-  AVAILABILITY_T  DRL    TABLE
-  AVAILABILITY_W  DRL    TABLE
-  CICS_A_BASIC_H  DRL    TABLE
-  CICS_A_BASIC_W  DRL    TABLE
-  CICS_A_DBCTL_H  DRL    TABLE
-  CICS_A_DBCTL_USR_H DRL    TABLE
-  CICS_A_DBCTL_USR_W DRL    TABLE
-  CICS_A_DBCTL_W  DRL    TABLE
-  CICS_A_DLI_H    DRL    TABLE
-  CICS_A_DLI_USR_H DRL    TABLE

Command ==>
F1=Help    F2=Split    F3=Exit    F5=Updates  F6=PurCond  F7=Bkwd
F8=Fwd     F9=Swap     F10=Actions F11=Display F12=Cancel
```

Figure 63. Tables window

The name of each table is shown in the **Tables** column.

The prefix of each table is shown in the **Prefix** column. Data tables and lookup tables have a prefix of DRL, the default value of the Prefix for all other tables field in the **Dialog Parameters** window. Control tables have a prefix of DRLSYS, the default value of the Prefix for system tables field in the **Dialog Parameters** window.

The **Type** column shows whether an object is a DB2 table or a view.

Working with data in tables

This section describes these tasks:

- Displaying the contents of a table
- Editing the contents of a table
- Showing the size of a table
- Recalculating the contents of a table
- Importing the contents of an IXF file to a table (This option is available only if your installation uses QMF with Tivoli Decision Support for z/OS.)
- Exporting table data to an IXF file (This option is available only if your installation uses QMF with Tivoli Decision Support for z/OS.)
- Purging a table
- Unloading and loading a table

Displaying the contents of a table

About this task

You can use the administration dialog to display the contents of a table.

Note: If QMF is not used with Tivoli Decision Support for z/OS on your system, the following applies:

- Tables are displayed with ISPF browse.

- The Add rows and Change rows options on the Edit pull-down are not selectable.
- If you display a very large table, data table, or system table, you might run out of REXX storage. If this happens, there are some actions you can take that enable you to display the table, or the part of the table you want to see.
 - Increase the region size.
 - If you need to see only the first part of the table, you can decrease the SQLMAX parameter on the Dialog Parameters window.
 - Use F4 (Run) on the SQL Query pop-up in the reporting dialog. Write an SQL SELECT statement that restricts the retrieved table information to the columns and rows you are interested in. This is a way to create and run a query without having to save it.

To display the contents of a table:

Procedure

1. From the Tables window, select the name of the table that you plan to display.
2. Press F11, or select 1, Display, from the Utilities pull-down. The product displays the contents of the table in rows and columns.

Note: The table is not necessarily sorted in key sequence.

REPORT	DRL.SAMPLE_H				LINE 1	POS 1	79
DATE	TIME	SYSTEM ID	DEPARTMENT NAME	USER ID	TRANSACTIONS	RESPONSE SECONDS	
2000-01-01	13.00.00	SYS1	Sys Supp	PIANKA	40	267	
2000-01-01	15.00.00	SYS1	Appl Dev	ADAMS	72	198	
2000-01-02	08.00.00	SYS1	Appl Dev	JONES	28	131	
2000-01-02	11.00.00	SYS1	Retail	PEREZ	21	171	
2000-01-03	10.00.00	SYS1	Marketng	KWAN	74	220	
2000-01-03	11.00.00	SYS1	Manufact	LEE	22	234	
2000-01-03	11.00.00	SYS1	Manufact	LUTZ	2	95	
2000-01-04	07.00.00	SYS1	Finance	HAAS	26	109	
2000-01-04	07.00.00	SYS1	Sys Supp	THOMPSON	84	64	
2000-01-04	08.00.00	SYS1	Marketng	KWAN	63	290	
2000-01-04	08.00.00	SYS1	Finance	GEYER	94	131	
2000-01-04	08.00.00	SYS1	Finance	GEYER	94	131	
2000-01-04	09.00.00	SYS1	Marketng	STERN	51	162	
2000-01-04	09.00.00	SYS1	Manufact	PULASKI	69	76	
1=Help		2=	3=End	4=Print	5=Chart	6=Query	
7=Backward		8=Forward	9=Form	10=Left	11=Right	12=	
OK, DRL.SAMPLE_H is displayed.							
COMMAND ===>					SCROLL ==> PAGE		

Figure 64. Using QMF to display a Tivoli Decision Support for z/OS table

3. Press F3 when you finish viewing the contents of the table, You are returned to the Tables window.

Editing the contents of a table

About this task

You can use the administration dialog to edit the contents of a table, using either the QMF table editor (if QMF is used with Tivoli Decision Support for z/OS) or the ISPF editor.

Working with tables and update definitions

The QMF table editor can be used in two modes, add and change. For a complete description, refer to the *Query Management Facility: Learner's Guide*

To add rows to a table using the QMF table editor:

Procedure

1. From the **Tables** window (Figure 63 on page 164), select the table to edit.
2. Select 1, Add rows, from the Edit pull-down.
The product calls the QMF table editor in add mode.
3. Enter values for columns, and press F2.
4. Press F3 when you finish adding rows.
QMF prompts you for confirmation.
5. Press Enter.
You are returned to the **Tables** window.

Changing or deleting rows using the QMF table editor

About this task

To change or delete rows using the QMF table editor:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table to edit
2. Select 2, Change rows, from the Edit pull-down.
Tivoli Decision Support for z/OS calls the QMF table editor in change mode.
3. To search for rows to change or delete, type values to search for, and press F2.
QMF displays the first row that matches the search criteria.
4. To change the row, type values for columns, and press F2.
5. To delete the row, press F11.
6. Press F3 when you finish changing or deleting rows.
QMF prompts you for confirmation.

Note: The ISPF edit function in the product administration dialog works according to ISPF rules. If no value is entered or if the value is removed, the character-type fields are filled with blanks. The ISPF Editor works the same way outside the dialog: that is, you can enter NULL values in Edit mode by typing HEX on the command line and X'00' in the field.

7. Press Enter.
You are returned to the Tables window.

What to do next

If all columns in a table row can be displayed in 32 760 characters (if you are using ISPF version 4 or later, otherwise 255 characters), you can use the ISPF editor to edit the table. If the table has more rows than the value you set for the SQLMAX value field in the Dialog Parameters window, TDS prompts you to temporarily override the default for this edit session.

Tivoli Decision Support for z/OS deletes all rows from the table and then reinserts them when you use this function. Because of this, the ISPF editor is not recommended for large tables.

Editing a table using the ISPF editor:**About this task**

To edit a table using the ISPF editor:

Procedure

1. From the **Tables** window (Figure 63 on page 164), select the table to edit
2. Select 3, ISPF editor, from the Edit pull-down.
3. Tivoli Decision Support for z/OS copies table rows to a sequential file and accesses the ISPF editor.

```

ISREDDE - STROMBK.DRLTAB ----- COLUMNS 001 017
***** ***** TOP OF DATA *****
==MSG> Use Tab Key to position to the next column
===== USER_ID | DEPARTME
=====         | NT_NAME
=====         | -----
000001 ADAMS     Appl Dev
000002 GEYER     Finance
000003 GOUNOT    Retail
000004 HAAS      Finance
000005 JONES     Appl Dev
000006 KWAN      Marketng
000007 LEE       Manufact
000008 LUTZ      Manufact
000009 MARINO    Retail
000010 MEHTA     Manufact
000011 PARKER    Finance
000012 PEREZ     Retail
000013 PIANKA    Sys Supp
000014 PULASKI   Manufact
000015 SMITH     Appl Dev
COMMAND ==>
                                SCROLL ==> CSR
F1=Help      F2=          F3=End      F4=          F5=R Find   F6=R Change
F7=Backward  F8=Forward   F9=         F10=Left    F11=Right  F12=Cursor

```

Figure 65. Editing a table in ISPF

4. Make any modifications to the table rows. You can add, delete, and change rows.
5. To cancel the changes, type CANCEL on the command line, and press Enter. You are returned to the **Tables** window without changing the table.
6. Press F3 when you finish editing the table.
The rows are reinserted into the DB2 table and you are returned to the Tables window.

Showing the size of a table**About this task**

Monitor the size of tables periodically to ensure that they are not getting too large.

Use the DB2 RUNSTATS utility to get information about tables and store it in the DB2 catalog each time you need current information about any DB2 database, including the Tivoli Decision Support for z/OS database. As described in “Monitoring the size of the Tivoli Decision Support for z/OS database” on page 108, Tivoli Decision Support for z/OS provides a sample job, DRLJRUNS, as an example of how to run the RUNSTATS utility. You can also run the RUNSTATS utility using these steps:

Working with tables and update definitions

Procedure

1. From the list of tables, select the Maintenance pull-down *without selecting a table*.
2. Select option 1, Tablespace.
3. From the list of table spaces, select one or more table spaces (or make no selection to process all the table spaces) and select the Utilities pull-down, as shown in Figure 31 on page 97.
4. Select option 2, Run DB2 RUNSTATS.

What to do next

To learn more about the DB2 RUNSTATS utility, refer to the *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

Use the administration dialog to check the size of tables in the product database:

1. From the Tables window (Figure 63 on page 164), select tables to display their sizes.

Note: If you do not select any tables, Tivoli Decision Support for z/OS displays the size of all tables in the product database.

2. Select 2, Show size, from the Utilities pull-down.

The Table Size window is displayed (Figure 66).

```
Table Maintenance Utilities Edit View Other Help
-----
Tables ROW 1 TO 13 OF 212
Table Size ROW 1 TO 10 OF 212
Press Enter to return.
Name Prefix Row Row length Kbytes
MVSPM_DEVICE_H DRL 80927 240 18967
MVSPM_DEVICE_AP_H DRL 34821 102 3468
MVSPM_CHANNEL_H DRL 9338 140 1276
MVSPM_APPL_H DRL 2388 491 1145
MVSPM_WORKLOAD_H DRL 2727 308 820
MVSPM_STORAGE_H DRL 2567 199 498
MVSPM_PAGE_DS_H DRL 966 229 216
MVSPM_XCF_PATH_H DRL 1296 171 216
MVSPM_SWAP_H DRL 1771 114 197
MVSPM_ENQUEUE_H DRL 1642 100 160
Command ==>
F1=Help F2=Split F7=Bkwd F8=Fwd F9=Swap F12=Cancel
F8=Fwd F9=Swap F10=Actions F11=Display F12=Cancel
```

Figure 66. Table Size window

Note:

- a. You can use the SORT command (for example, SORT KBYTES DESC) to find the largest tables.
 - b. If the information shown in the Table Size window is incomplete, run the DB2 RUNSTATS utility and restart this procedure.
3. After you finish viewing this window, press Enter.
You are returned to the Tables window.

Recalculating the contents of a table

About this task

Sometimes tables get filled with incorrect data during the collect process. (This can be caused by a situation such as a bad record in a log.) For a single, independent table, you can correct these problems using one of the options on the Edit pull-down. Tivoli Decision Support for z/OS provides a recalculate function for the following special conditions:

- When tables are updated from other tables and corrections must be propagated to all dependent tables.
- When a key column is changed to a new value, and data already exists for the new key.

You can also use the recalculate function to populate a new table from another table, for example a monthly table from a daily table.

You can use the administration dialog to recalculate the contents of tables. For more information about the RECALCULATE log collector language statement, refer to the *Language Guide and Reference*.

To recalculate the contents of tables:

Procedure

1. From the Tables window (Figure 63 on page 164), select the source table (the table you plan to modify)
2. Select 8, Recalculate, from the Utilities pull-down.
The Recalculate window is displayed (Figure 67).

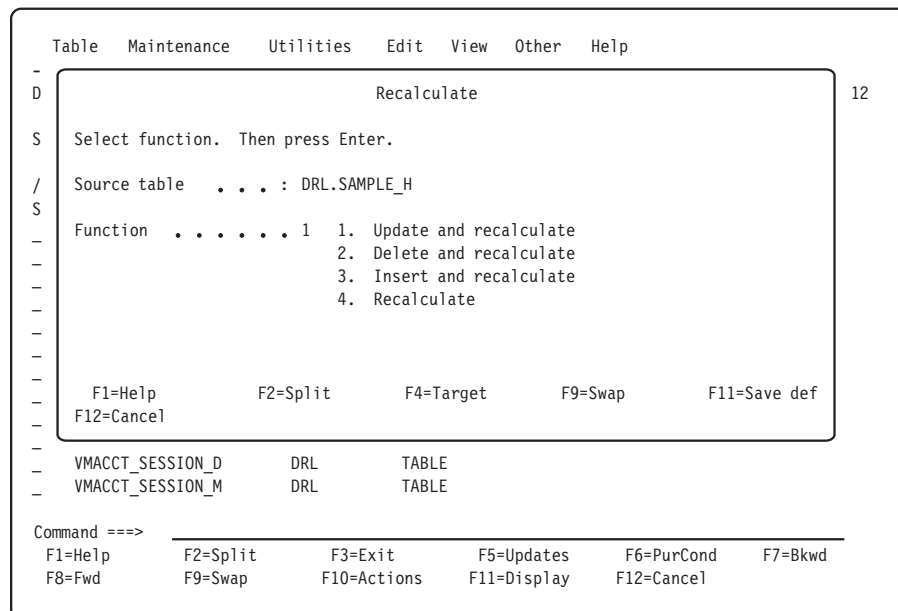


Figure 67. Recalculate window

3. Optionally, press F4 to specify target tables (the tables that changes in the source table should be propagated to). If you do not specify target tables, changes are propagated to all affected tables.

The Target Tables window is displayed.

Working with tables and update definitions

4. Select one or more target tables from the list and press Enter.
You are returned to the Recalculate window.
5. Select the desired function from the list and press Enter. Options 1, 2, and 3 are used to modify the source table. Option 4 propagates selected source table rows without changing the source table.

If you did not choose to insert and recalculate (option 3), the Condition window is displayed (Figure 68).

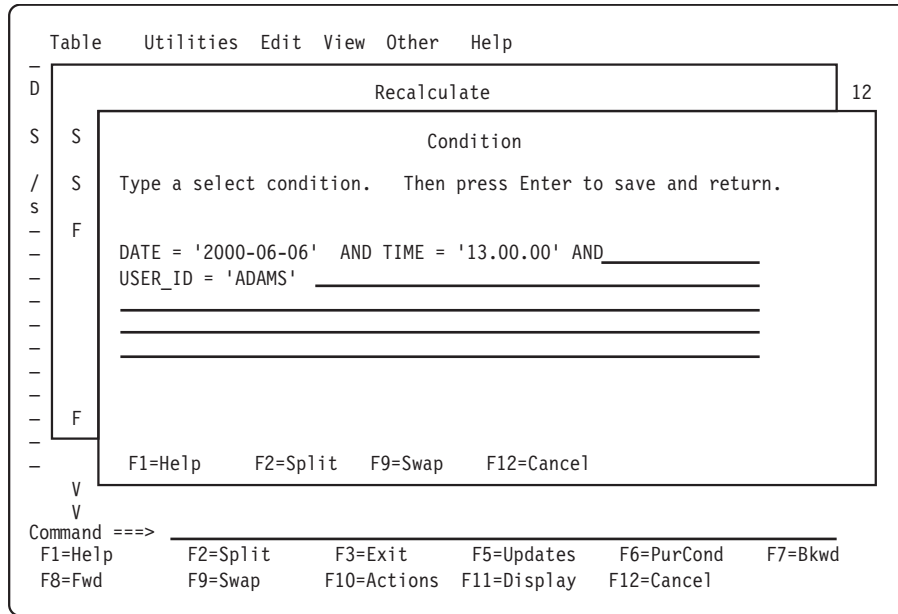


Figure 68. Condition window

6. Specify a condition to restrict rows affected in the source table and press Enter.
If you choose to update and recalculate (option 1) or insert and recalculate (option 3), the Column Values window is displayed (Figure 69 on page 171).

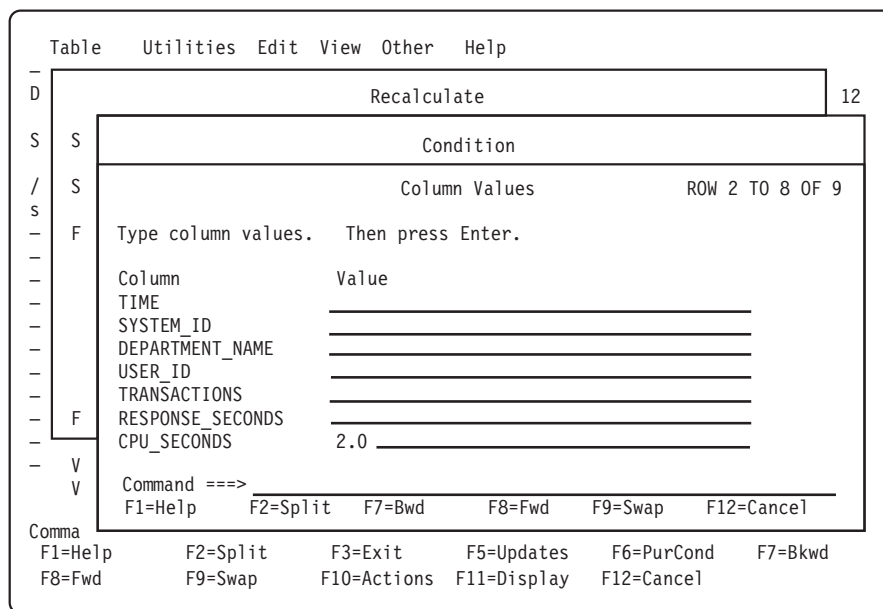


Figure 69. Column Values window

7. Type column values in the fields, and press Enter.
The recalculate function is performed and you are returned to the Recalculate window.
8. Press F12 to return to the Tables window.

Importing the contents of an IXF file to a table

About this task

You might want to import data from another source into a Tivoli Decision Support for z/OS table. If QMF is used with Tivoli Decision Support for z/OS, you can use the administration dialog to import data in the Integration Exchange Format (IXF). Refer to the *QMF Application Development Guide* for a description of the IXF format.

Note: When you import the file, Tivoli Decision Support for z/OS replaces the contents of the table.

To import data into a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table.
2. Select 3, Import, from the Utilities pull-down.
The Import Data Set window is displayed.
3. Type the name of the data set that contains the data you want to import and press Enter.
The data is imported into the table and you are returned to the Tables window.

Exporting table data to an IXF file

About this task

You might want to export data from a Tivoli Decision Support for z/OS table to an IXF data set. If QMF is used with the product, you can use the administration dialog to do this.

To export data from a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table.
2. Select 4, Export, from the Utilities pull-down
The Export Data Set window is displayed.
3. Type the name of the data set to export data into, and press Enter.
The data is exported into the data set you specified and you are returned to the Tables window.

Purging a table

About this task

Each table in the product database is associated with a purge condition that determines how long the data in the table is kept. See “Displaying and editing the purge condition of a table” on page 190 for a description of how to define the purge condition for a table.

Purging the database is normally a batch process. See “Purge utility” on page 104 for a description of how to run purge in batch.

You can also use the administration dialog to delete the data specified by the purge condition:

Procedure

1. From the Tables window (Figure 63 on page 164), select tables to purge.

Note: If you do not select any tables, Tivoli Decision Support for z/OS purges the contents of all data tables with purge conditions.
2. Select 9, Purge, from the Utilities pull-down.
The Purge Confirmation window is displayed.
3. Press Enter to confirm the purge.
The purge conditions associated with the tables are run and the statistics on the number of rows deleted from each table are displayed.

Unloading and loading tables

About this task

When you need to change a DB2 table, for example by adding a column, you can save the existing data by using the DB2 Unload utility. After the change to the table, you then reload the table using the Load utility. Using Unload and Load with no change reorganizes the data.

Moreover, the possibility of reading and writing a data set of data directly on tape improves possible recovery and backup operations.

The Load utility is used to load data into a table of a table space. It enables you to load records into the tables and builds or extends any indexes defined on them. If the table space already contains data, you can either add the new data, or replace the existing data with the new data. Because the Load utility operates at a table space level, to run it you must have the required authority for all the tables of the table space. The data set used for the Load utility can be read from both disk and tape. The Unload utility is used to unload data from a table to a sequential data set. To use the Unload utility, the definitions of the table space and tables must be available on the system. The data set used for the unload operation can be saved both on disk and tape.

Note: Load and Unload work only with tables, and cannot be used with views.

To unload the contents of a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the tables to unload, as shown in Figure 70.

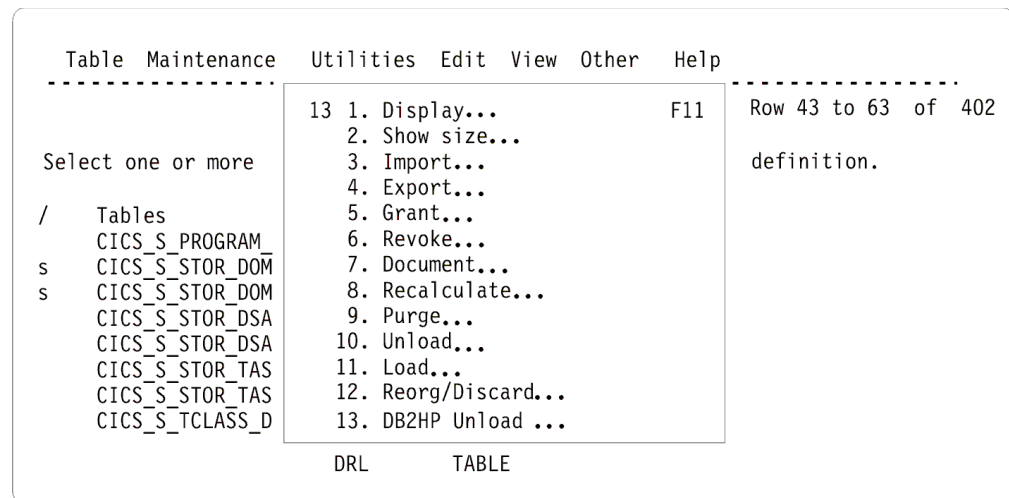


Figure 70. Selecting tables to unload

2. Select option Unload, from the Utilities pull-down menu.
The Unload Utility window opens, as shown in the following figure:

Working with tables and update definitions

```

Table  Maintenance  Utilities  Edit  View  Other  Help

                                UNLOAD Utility

The Unload utility will unload table data to a data set. Type the fully
qualified data set name, without quotes. Then press Enter to create the
JCL.

Type of UNLOAD.....1  1.  Disk
                               2.  Tape
Table.....: AVAILABILITY_D
UNLOAD data set name..... SAMPLE.DAT

Type information in the following fields. In case of Tape UNLOAD,
VOLSER is the tape label. In case of Disk UNLOAD, type information
only if the data set is not available.
UNIT..... _____
VOLSER..... _____

F1=Help  F2=Split  F9=Swap  F12=Cancel

F8=Fwd   F9=Swap   F10=Actions  F11=Display  F12=Cancel
    
```

Figure 71. Unload Utility window

3. From the Unload Utility window, specify the unload type by inserting **1** for disk unload or **2** for tape unload. The default is Disk Unload.
4. Specify the name of the table and data set you want to unload.
5. If you selected Disk Unload:

Option	Description
If the data set already exists	Leave the fields UNIT and VOLSER blank. If you need to create a new data set, enter the required information in both the fields.
If you selected Tape Unload	Specify the tape unit in the UNIT field, and the tape label in the VOLSER field.

6. When you are finished, press Enter.
A JCL is created and saved in your library, so that it can be used later. When the JCL is launched two data sets are automatically created. One is used to reload data (SYSPUNCH) and the other contains the data unloaded by the utility.

Note: When using Load on a multiple table space, you must be careful because Load works on a complete table space at a time (for information about replacing data with Load, refer to the *DB2 for OS/390 V5 Utility Guide and Reference*). This applies especially when tables are dropped and recreated.

For this reason, when you apply PTFs involving tables that need to be dropped and recreated, you should follow these steps:

- a. Unload the tables, if you want to keep the previously collected data.
- b. Use SMP/E to apply the PTF.
- c. Execute the SQL drop table statement of the above tables using either of the following:
 - DB2 SPUFI

- Option 5, Process Tivoli Decision Support for OS/390 statements, from the Other pull-down on any primary window of the Tivoli Decision Support for OS/390 administration dialogs.
- d. Execute the SQL create table statements for the same tables using either of the following methods:
- Reinstall the component.
 - Select Option 5, Process Tivoli Decision Support for OS/390 statements, from the Other pull-down on any primary window of the Tivoli Decision Support for OS/390 administration dialogs. Execute the definition members of the local or the standard definition library, depending on whether or not the definitions have been user-modified. Ignore the error messages issued for the existing objects and make sure that the changed tables are correctly created.
- e. Load your previously unloaded data.

What to do next

To generate a job that reloads the data, from the Tables window, select option 11, Load. Then enter the required information, as explained above.

The following example shows control statements for the Unload utility. Data is unloaded from the AVAILABILITY_D table onto tape. The DDNAME for SYSPUNCH data set is completed with the UNIT and VOLSER information about the Tape Unit used. The data set input from panel is SYSREC00.

```
//UNLOAD JOB (ACCOUNT),'NAME'
/**
/** THIS JCL HAS BEEN REWRITTEN IN ORDER
/** TO PROPERLY UNLOAD THE DATA FROM DB2 TABLES.
/** DSNTIAUL IS USED FOR UNLOAD INSTEAD OF DSNUPROC
/** UTILITY.
/** THEREFORE, PLEASE, NOTE THAT THIS IS ONLY
/** A SAMPLE THAT NEEDS TO BE PROPERLY CUSTOMIZED.
/** WARNINGS :
/** PLEASE CHECK PLAN NAME (NORMALLY DSNTIBVR),
/** V=DB2 VERSION, AND R=DB2 RELEASE;
/** TWO NEW DATASETS ARE DEFINED (SYSREC00 AND SYSPUNCH).
/** SYSPUNCH DATASET, IS CREATED AT UNLOAD STEP,
/** as USERID.SYSPUNCH (USERID.SYSPUNCH).
/** SYSREC00 DATASET IS SELECTED FROM the PREVIOUS PANEL.
/**
/** I M P O R T A N T :
/** CHECK THE DATA SET PARAMETER IF YOU HAVE CHOSEN
/** THE UNLOAD ON TAPE.
/**
//UNLOAD EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DSN6)
RUN PROGRAM(DSNTIAUL) PLAN(DSNTIB71) -
PARMS('SQL') LIB('DSN710.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSREC00 DD DSN=USERID.DAT.UNLOAD,
// UNIT=TAPE_UNIT,
// SPACE=(4096,(5040,504)),
// DISP=(,PASS),
// LABEL=(1,SL),
// DCB=(RECFM=FB,LRECL=410,BLKSIZE=27880),
// VOL=SER=TAPE_LABEL
//SYSPUNCH DD DSN=USERID.SYSPUNCH,
// UNIT=xxxx,
```

Working with tables and update definitions

```
//          VOL=SER=xxxxxx,  
//          SPACE=(4096,(5040,504)),  
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920),  
//          DISP=(NEW,CATLG,CATLG)  
//SYSIN    DD *  
           SELECT * FROM USERDB.AVAILABILITY_D;
```

The following example shows control statements for the Load utility. Data is loaded from tape into the AVAILABILITY_D table. The DDNAME for the SYSPUNCH data set is completed with the UNIT and VOLSER information about the Tape Unit used. The data set input from panel is SYSREC00.

```
//LOAD JOB (ACCOUNT),'NAME'  
//*  
//* THIS JCL HAS BEEN REWRITTEN IN ORDER  
//* TO PROPERLY LOAD THE DATA FROM DB2 TABLES.  
//* DSNTIAUL IS PREVIOUSLY USED FOR UNLOAD  
//* INSTEAD OF DSNUPROC UTILITY.  
//* THEREFORE, PLEASE, NOTE THAT THIS IS ONLY  
//* A SAMPLE THAT NEEDS TO BE PROPERLY CUSTOMIZED.  
//* WARNINGS :  
//* PLEASE CHECK PLAN NAME (NORMALLY DSNTIBVR),  
//* V=DB2 VERSION, AND R=DB2 RELEASE;  
//* TWO NEW DATASETS ARE DEFINED (SYSREC00 AND SYSPUNCH).  
//* as USERID.SYSPUNCH (USERID.SYSPUNCH).  
//* SYSREC00 DATASET IS SELECTED FROM the PREVIOUS PANEL  
//*  
//*  
//*           I M P O R T A N T :  
//* SYSPUNCH DATASET NEEDS TO BE EDITED FROM USER  
//* BEFORE EXECUTING LOAD,  
//* INSERTING "RESUME YES LOG YES" OPTIONS,  
//* IN ORDER TO CONTAIN COMMAND :  
//* "LOAD DATA RESUME YES LOG YES INDDN  
//* SYSREC00 INTO TABLE tablename"  
//* CHECK THE DATA SET PARAMETER IF YOU HAVE CHOSEN  
//* THE LOAD FROM TAPE.  
//*  
//LOAD EXEC DSNUPROC,PARM='DSN6,MYUID'  
//DSNTRACE DD SYSOUT=*  
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR  
//SORTWK01 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SORTWK02 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SORTWK03 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SORTWK04 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SORTOUT DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SYSREC00 DD DSN=USERID.DAT.UNLOAD,  
//          UNIT=TAPE UNIT,VOL=SER=TAPE_LABEL,  
//          LABEL=(1,SL),  
//          DISP=SHR  
//SYSUT1 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND)  
//SYSIN DD DSN=USERID.SYSPUNCH,DISP=SHR
```

Integration with DB2 High Performance Unload

The DB2 High Performance Unload is a high-speed utility for unloading DB2 tables from either a table space or an image copy. Tables are unloaded to one or more files based on a specified format. You can use it to extract data for movement across enterprise systems or for reorganization in-place. DB2 HP Unload can do the following:

- Rapidly unload table spaces
- Run parallel unloads accessing the same table space
- Unload against any image copy to eliminate interference with DB2 production databases

- Unload selected rows and columns
- Unload a maximum number of rows, unloading one row out of every n rows
- Generate load control statements for a subsequent reload.

The DB2 High Performance Unload can manage an UNLOAD command and an optional SELECT statement. The syntax of the SELECT statement is compatible with the syntax of the DB2 SELECT statement. The SELECT statement is used to define which table data must be extracted onto data set or tape (for example, if in your table a DATE field is present, you can extract all the data with a date later than 2002-01-01, by writing the appropriate WHERE condition in the SELECT statement of the UNLOAD command).

Running DB2 High Performance Unload utility

About this task

To run the DB2 High Performance Unload utility, you must have the product correctly installed and configured on the system.

Note: The DB2HP Unload utility integration works in batch mode; it can run in interactive mode only if you have DB2 Administration Tool, or DB2 Tools Launchpad, installed on your system. These products are optional and not needed to run the DB2HP Unload utility.

To run the utility follow these steps:

Procedure

1. From the Tables window, select the table to unload, as shown in Figure 63 on page 164
2. From the Utilities pull-down menu, select option DB2HP Unload, as shown in Figure 70 on page 173.

Note: The DB2 High Performance Unload utility can only be run on tables. It cannot be run on views.

3. From the DB2 High Performance Unload Utility window, specify the unload type by inserting **1** for disk unload or **2** for tape unload. The default value is disk unload. Then, specify the name of data set that will be used to store the unloaded data, as shown in the following window.

```

Table  Maintenance  Utilities  Edit  View  Other  Help

                                DB2HP Unload Utility

The DB2HP Unload utility will unload table data to a data set. You can use the
utility only if the DB2HP product is present on system. Type the fully
qualified data set name, without quotes. Then press Enter to create the
JCL.

Type of DB2HP Unload.....1 1.  Disk
                               2.  Tape
Table.....: AVAILABILITY_D
Unload data set name..... SAMPLE.DAT

Type information in the following fields. In case of Tape UNLOAD, VOLSER
is the tape label. In case of Disk UNLOAD, type information only if the
data set is not available.
UNIT..... _____
VOLSER..... _____

F1=Help  F2=Split  F9=Swap  F12=Cancel

F8=Fwd   F9=Swap   F10=Actions  F11=Display  F12=Cancel
    
```

Figure 72. DB2 High Performance Unload utility

4. If you selected Disk:

- If the data set already exists, leave the fields UNIT and VOLSER blank. If you need to create a new data set, enter the required information in both the fields.

If you selected Tape:

- Specify the tape unit in the UNIT field, and the tape label in the VOLSER field

5. When you are finished, press Enter.

A JCL is created and saved in your library so that it can be used later. When the JCL is launched two data sets are automatically created. One is used to reload data (SYSPUNCH), the other contains the data unloaded by the utility.

Sample control statement for DB2 High Performance Unload utility:

Data has been unloaded from the AVAILABILITY_D table; the DDNAME for SYSPUNCH data set must be completed with UNIT and VOLSER information. The data set input from panel is SYSREC00.

```

//DB2HPU JOB (ACCOUNT),'NAME'
/**
/** THIS JCL HAS BEEN REWRITTEN IN ORDER
/** TO PROPERLY UNLOAD THE DATA FROM DB2 TABLES.
/** THE DB2 High Performance Unload (INZUTILB)
/** IS USED FOR UNLOAD DATA IN BATCH MODE.
/** THEREFORE, PLEASE, NOTE THAT THIS IS ONLY
/** A SAMPLE THAT NEEDS TO BE PROPERLY CUSTOMIZED.
/** WARNINGS :
/** V=DB2 VERSION, AND R=DB2 RELEASE;
/** TWO NEW DATASETS ARE DEFINED (SYSREC00 AND SYSPUNCH).
/** SYSPUNCH DATASET, IS CREATED AT UNLOAD STEP,
/** as USERID.SYSPUNCH (USERID.SYSPUNCH).
/** SYSREC00 DATASET IS SELECTED FROM the PREVIOUS PANEL.
/**
/** I M P O R T A N T :
/** CHECK THE DATA SET PARAMETER IF YOU HAVE CHOSEN
/** THE UNLOAD ON TAPE.
/**
    
```

```
//STEP1 EXEC PGM=INZUTILB,REGION=0M,DYNAMNBR=99,
//      PARM='DSN6,DB2UNLOAD'
//STEPLIB DD DSN=DSN710.SINZLINK,DISP=SHR
//*
//SYSIN DD *
        UNLOAD TABLESPACE PRM1DB.DRLSCOM
        DB2 YES
        QUIESCE YES QUIESCECAT YES
        OPTIONS DATE DATE_A
        SELECT * FROM PRM1.AVAILABILITY_D
        OUTDDN (SYSREC00)
        FORMAT DSNTIAUL
        LOADDDN SYSPUNCH LOADOPT (RESUME NO REPLACE)
/*
//SYSPRINT DD SYSOUT=*
//*
//***** DDNAMES USED BY THE SELECT STATEMENTS *****
//*
//SYSREC00 DD DSN=SAMPLE.DAT,
//          UNIT=3390,
//          SPACE=(4096,(1,1)),
//          DISP=(NEW,CATLG,CATLG),
//          DCB=(RECFM=FB,LRECL=410,BLKSIZE=27880),
//          VOL=SER=MYVOL
//SYSPUNCH DD DSN=USERID.SYSPUNCH,
//          UNIT=xxxx,
//          VOL=SER=xxxxxx,
//          SPACE=(4096,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920),
//          DISP=(NEW,CATLG,CATLG)
```

Working with tables and update definitions

The rest of this chapter describes working with tables and update definitions.

Opening a table to display columns

About this task

You can use the administration dialog to view a table definition.

To open a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table definition you plan to view.
2. Press Enter.

The table definition is opened. Figure 73 on page 180 shows an example of an opened table definition.

Working with tables and update definitions

```
Table SAMPLE_H                                ROW 1 TO 9 OF 9

Select a column. Then press Enter to display the definition.

Database . . . : DRLDB                        Tablespace . . . : DRLSSAMP
Comments . . . : This table provides hourly sample data. >
/ Column      Type      Length Nulls Primary Key
- DATE        DATE        4      No   Yes
- TIME        TIME        3      No   Yes
- SYSTEM_ID   CHAR         4      No   Yes
- DEPARTMENT_NAME CHAR       8      No   Yes
- USER_ID     CHAR         8      No   Yes
- TRANSACTIONS INTEGER      4      Yes  No
- RESPONSE_SECONDS INTEGER     4      Yes  No
- CPU_SECONDS  FLOAT       8      Yes  No
- PAGES_PRINTED INTEGER     4      Yes  No
***** BOTTOM OF DATA *****

Command ==>
F1=Help      F2=Split    F3=Exit     F5=Add col  F6=Indexes  F7=Bkwd
F8=Fwd       F9=Swap     F10=Show fld F12=Cancel
```

Figure 73. Table window

3. Type changes to comments in the Comments field and press Enter.

Note: Press F10 to see the entire Comments field.

The changes to the comments are saved.

Displaying and modifying a column definition About this task

To display and modify a column definition:

Procedure

1. From the Table window, select the column, and press Enter.

The Column Definition window for the column is displayed (Figure 74 on page 181).

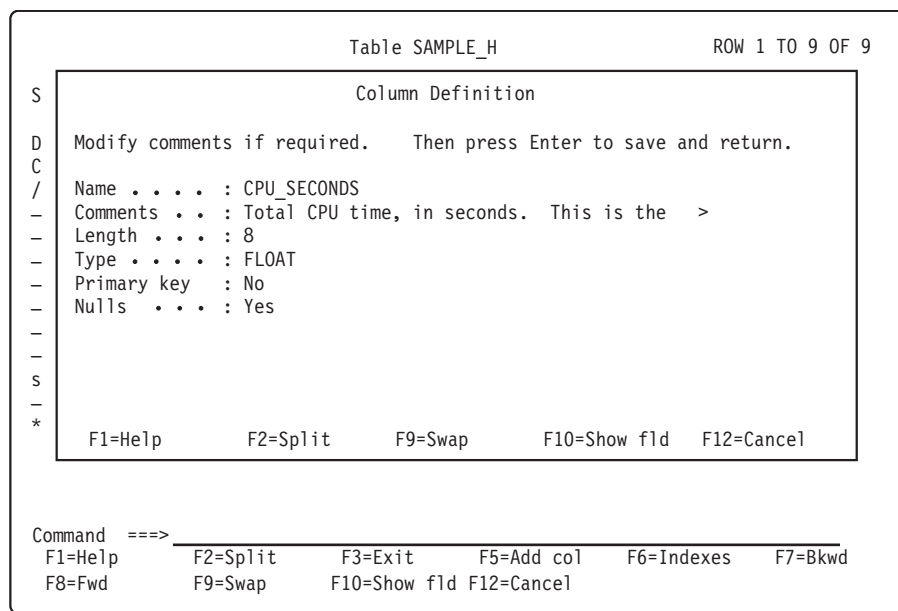


Figure 74. Column Definition window

2. Type changes to comments in the Comments field, and press Enter.

Note: Press F10 to see the entire Comments field.

The changes are saved and you are returned to the Tables window.

Adding a column to a table

About this task

You can add columns to a table, but you cannot delete columns.

To add a column to a table:

Procedure

1. From the Table window, press F5.

The Add Column window is displayed(Figure 75 on page 182).

Working with tables and update definitions

```

-
S
-
Add Column
-
Type column information.  Then press Enter to save and return.
D
C
/
Name . . . _____ (required)
Comments . . . _____ >
-
Length . . . _____ Primary key . 2 1. Yes
-
-
-
Type . 1. Char Nulls . 1 1. Default
-
-
-
2. Varchar 2. NOT NULL
-
-
-
3. Smallint 3. NOT NULL WITH
-
-
-
4. Integer DEFAULT
-
-
-
5. Float
-
-
-
6. Decimal
-
-
-
7. Date
-
*
-
8. Time
-
9. Timestamp
-
10. Graphic
-
11. Vargraphic
-
12. Long varchar
-
13. Long vargraphic
C
F1=Help F2=Split F9=Swap F10=Show fld F12=Cancel

```

Figure 75. Add Column window

2. Type information for the new column in the window, and press Enter.
The new column is added to the table and you are returned to the Add Column window.
3. When you finish adding columns to the table, press F12.
You are returned to the Tables window.

Displaying and adding a table index About this task

If a table has a primary key, it must have an index on that key (the primary index). Some queries access tables using the primary index.

A table can have more than one index. Secondary indexes can give you faster data retrieval, but increase the amount of time that collect requires to update those tables.

Note: If you want to work with index **spaces**, see “Displaying and modifying a table or index space” on page 192.

To view or add indexes to a table:

Procedure

1. From the Tables window, select a table and press Enter.
2. From the Table window, press F6.
The Indexes window is displayed (Figure 76 on page 183).

```

                                     Indexes                                ROW 1 FROM 1
Select an index. Then press Enter to display.

/  Indexes          Table          Unique Cluster
_  SAMPH_IX        SAMPLE_H        Yes     Yes
***** BOTTOM OF DATA *****

Command ==> _____
F1=Help   F2=Split   F3=Exit   F5=Add ind  F7=Bkwd   F8=Fwd
F9=Swap   F11=Delete  F12=Cancel
    
```

Figure 76. Indexes window

3. To view an index definition, select the index and press Enter. The Index window is displayed (Figure 77). The index on the primary key should be a unique, clustering index. Refer to the DB2 documentation for a description of the other index options.

```

                                     Index SAMPH_IX                        ROW 1 TO 7 OF 9
S
Press Enter to return.
/
/ Table name . . . . . : SAMPLE_H
*
Storage group . . . . . : SYSDEFLT          Subpages . . . :4
Primary quantity . . . . : 6                Unique . . . :YES
Secondary quantity . . . . : 3              Cluster . . . :YES
Erase . . . . . : NO                       Buffer pool . . :BPO

Column name      Column type      Seq  Order
DATE             DATE             1   ASC
TIME             TIME             2   ASC
SYSTEM_ID        CHAR             3   ASC
DEPARTMENT_NAME CHAR             4   ASC
USER_ID          CHAR             5   ASC
TRANSACTIONS     INTEGER
RESPONSE_SECONDS INTEGER

C Command ==> _____
  F1=Help   F2=Split   F7=Bkwd   F8=Fwd   F9=Swap   F12=Cancel
    
```

Figure 77. Index window

4. Press Enter to return to the Indexes window.
5. From the Indexes window, press F5 to add an index to the table. The Add Index window is displayed (Figure 78 on page 184).

Working with tables and update definitions

```

Add Index                                ROW 1 TO 5 OF 9
Type information.  Then press Enter to save and return.
Index name . . . . . _____ (required)
Table Name . . . . . : SAMPLE_H
Storage group . . . . . _____ Subpages . . . . . _____
Primary quantity . . . . . _____ Unique . . . . . 2 1. Yes
Secondary quantity . . . . . _____ 2. No
Erase . . . . . 1 1. No Cluster . . . . . 2 1. Yes
2. Yes 2. No
Bufferpool . . . . . BPO
Column name      Column type      Seq  Order
DATE             DATE             ___  ___
TIME             TIME             ___  ___
SYSTEM_ID        CHAR             ___  ___
DEPARTMENT_NAME CHAR             ___  ___
USER_ID          CHAR             ___  ___
Command ==> _____
F1=Help  F2=Split  F7=Bkwd  F8=Fwd  F9=Swap  F12=Cancel
  
```

Figure 78. Add Index window

- Type the information for the new index and press Enter.
The index is added to the table and you are returned to the Indexes window.

What to do next

Note: To modify an index, delete and recreate it.

Deleting a table index

About this task

To delete a table index:

Procedure

- From the Indexes window, select the index and press F11.
A confirmation window is displayed.
- Press Enter to confirm the deletion.
You are returned to the Indexes window.

Displaying and modifying update definitions of a table

About this task

The instructions for entering data from logs into DB2 tables in the product database are provided by update definitions. An *update definition* describes how the data in a source (a record or a table) is summarized into a target table during the collect process. Refer to the *Language Guide and Reference* for information about how to define update definitions using the log collector language.

Update definitions are supplied for all data tables. You can use the administration dialog to modify these update definitions.

To display and edit the update definitions of a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table and press F5. The Update Definitions window for the table is displayed (Figure 79). All update definitions where the selected table is either the source or the target are included.

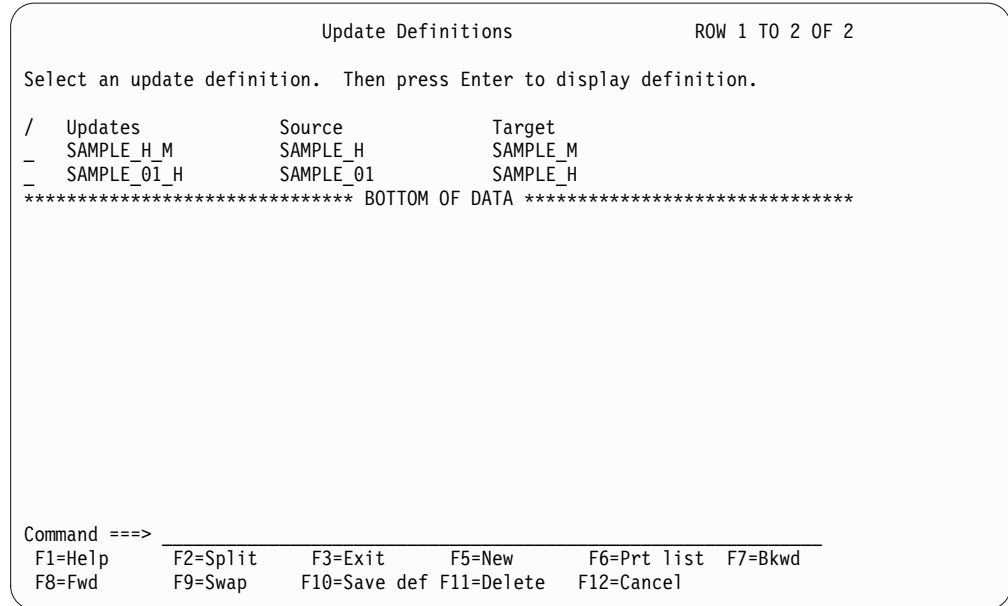


Figure 79. Update Definitions window

2. Select the update definition to modify and press Enter. The Update Definition window for the update definition is displayed (Figure 80).

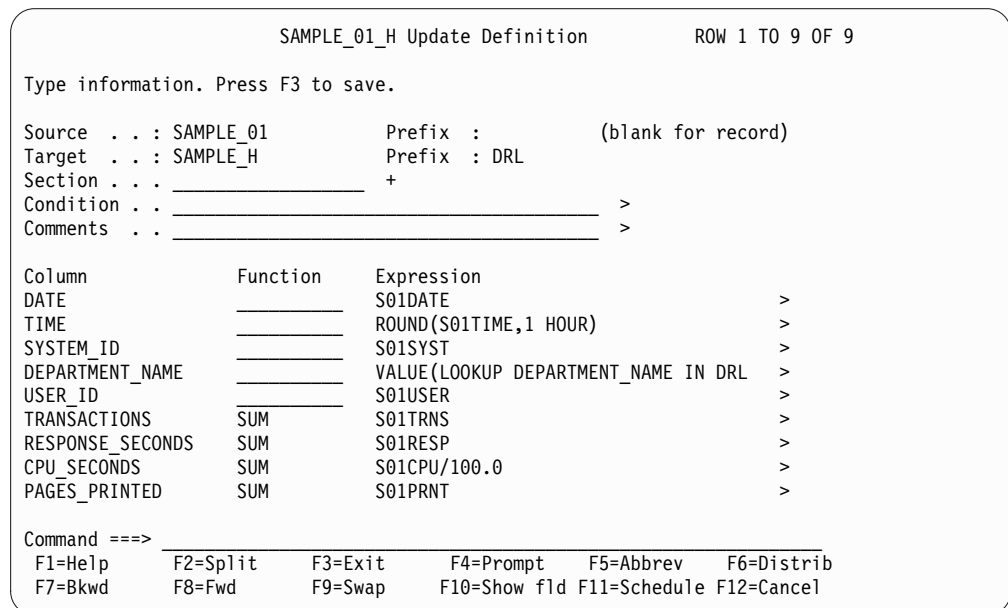


Figure 80. Update Definition window

Complete these fields in the window:

Working with tables and update definitions

Section

The name of a repeated section in a source record.

If the source is a record, you can type the name of a repeated section in this field. Tivoli Decision Support for z/OS uses the update during collection to process each repeated section.

Condition

A condition that is applied to source fields or columns.

Type an expression that evaluates as either true or false in this field. Tivoli Decision Support for z/OS evaluates the expression to determine if it is true before processing the source with the update.

Comments

A description of the update definition.

Column

All columns of the target table.

Function

Describes the accumulation function to use. Blank means that the column is a key (a GROUP BY column). For data columns, the value of this field can be SUM, MIN, MAX, COUNT, FIRST, LAST, AVG, and PERCENT.

To use the MERGE function, identify input to the function by designating a column for each of these functions: INTTYPE, START, END, and QUIET.

Expression

Describes how the value in the column should be derived from source fields, columns, or abbreviated names of expressions. (See “Working with abbreviations” on page 187 for more information.) If the update does not affect the value of the column, there is no entry in the expression field.

For an AVG column, type the expression, followed by a comma, and a column name. For a PERCENT column, type the expression, followed by a comma, a column name, a comma, and a percentile value (without the percent sign).

Refer to the *Language Guide and Reference* for more information about using log collector language:

- Functions
- Accumulation functions
- Expressions
- Statements
- Averages
- Percentiles

3. Type any modifications to the update definition in the fields.
4. Press F5 to modify abbreviations in this update definition.
The Abbreviations window is displayed. See “Working with abbreviations” on page 187, for more information.
5. Press F6 to modify the distribution clause associated with the update definition.
The Distribution window is displayed. See “Modifying a distribution clause” on page 188 for more information.

6. Press F11 to modify the apply schedule clause associated with an update definition.

The Apply Schedule window is displayed. See “Modifying an apply schedule clause” on page 189 for more information.

7. Press F3 when you finish modifying the update definition.

The changes are saved and you are returned to the Update Definitions window.

8. Repeat this procedure to modify other update definitions or press F3 again to return to the Tables window.

Working with abbreviations

About this task

You can use abbreviations to give names to long expressions that are used several times. Using abbreviations improves product performance because expressions are evaluated only once.

Defining abbreviations with the administration dialog is equivalent to using the LET clause in a log collector DEFINE UPDATE statement to assign an expression to a variable name. (Refer to the description of the DEFINE UPDATE statement in the *Language Guide and Reference* for more information.)

To modify an abbreviation:

Procedure

1. From the Update Definition window (Figure 80 on page 185), press F5.
The Abbreviations window is displayed (Figure 81).

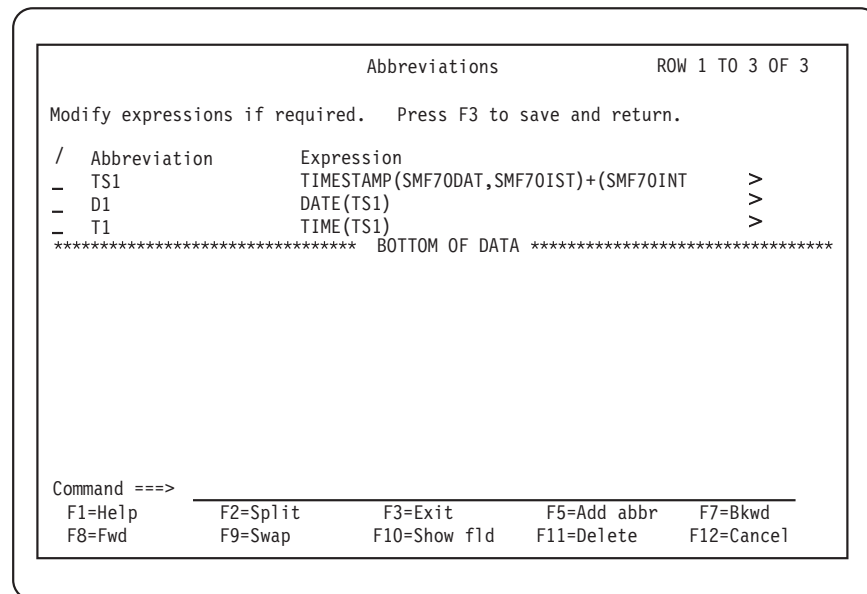


Figure 81. Abbreviations window

2. Type modifications in the fields and press Enter.

The changes are saved and you are returned to the Update Definition window.

Adding an abbreviation to an update definition:

Working with tables and update definitions

About this task

To add an abbreviation to an update definition:

Procedure

1. From the Abbreviations window, press F5.
The Abbreviation window is displayed.
2. Type the abbreviation and the expression in the fields and press Enter.
The abbreviation is added and you are returned to the Abbreviations window.

Deleting an abbreviation from an update definition:

About this task

To delete an abbreviation from an update definition:

Procedure

From the Abbreviations window, select the abbreviation to delete, and press F11.
The abbreviation is deleted from the list.

Modifying a distribution clause

About this task

The distribution clause of an update definition specifies that source fields or columns are distributed over a time period. It can be used when you have a record that contains data for a long time period and you do not want all values to be summarized at the start or end time.

To modify the distribution clause associated with an update definition:

Procedure

1. From the Update Definition window (Figure 80 on page 185), press F6.
The Distribution window is displayed (Figure 82 on page 189).

```

                                Distribution                                ROW 1 TO 7 OF 65

Type information.  Then press Enter to save and return.

By period . . . . . 60 * VALUE(LOOKUP TIME_RESOLUTION IN DRL.M > (seconds)
Start timestamp . . . . . TIMESTAMP(SMF33TSD,SMF33TST) >
End timestamp . . . . . TIMESTAMP(SMF33TED,SMF33TET) >
Timestamp . . . . . INTERVAL_START (any ID)
Interval . . . . . INTERVAL_LENGTH (any ID)

/ Column/Field
- SMF33ACL
- SMF33ACT
- SMF33ALN
- SMF33AOF
- SMF33AON
/ SMF33CN
/ SMF33CNA

Command ==>
F1=Help      F2=Split    F7=Bkwd     F8=Fwd      F9=Swap
F10=Show fld F11=Delete  F12=Cancel

```

Figure 82. Distribution window

2. Type modifications in the fields and press Enter.

The changes are saved and you are returned to the Update Definition window.

Modifying an apply schedule clause

About this task

Tivoli Decision Support for z/OS uses the apply schedule clause of an update definition in calculating availability. The clause specifies how the product should merge schedule information in control tables (see “Control tables” on page 243) with detailed availability information.

To modify the apply schedule clause associated with an update definition:

Procedure

1. From the Update Definition window (Figure 80 on page 185), press F11.
The Apply Schedule window is displayed (Figure 83 on page 190).

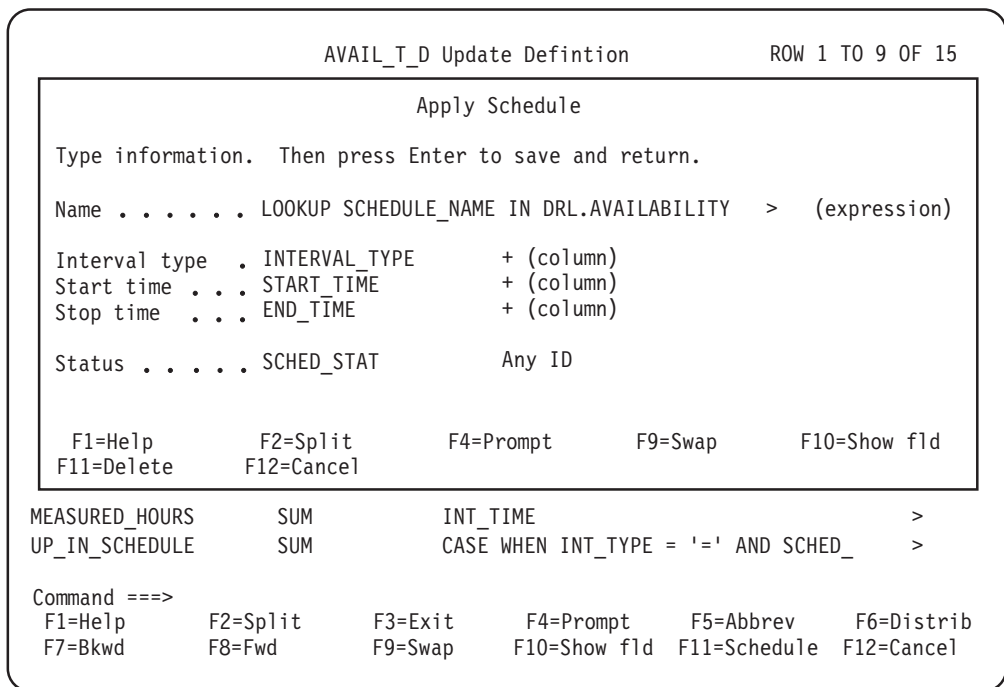


Figure 83. Apply Schedule window

2. Type modifications in the fields and press Enter.
 The changes are saved and you are returned to the Update Definition window.

What to do next

Refer to the *Language Guide and Reference* for more information about using the log collector language to:

- Determine resource availability
- Calculate the actual availability of a resource
- Compare actual availability to scheduled availability

Displaying and editing the purge condition of a table

About this task

Tivoli Decision Support for z/OS uses purge conditions to specify when old data should be purged from tables. A table can have only one purge condition. Purge conditions are supplied for all data tables. You can use the administration dialog to modify the purge condition of a table.

The administrative report PRA003 produces a complete list of all current product purge definitions. For more information about this report, see “PRA003 - Table purge condition” on page 284.

To display and edit the purge condition of a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table to update and press F6.

The Retention Period window is displayed (Figure 84) if the purge condition is blank or has the standard format (`column_name < CURRENT DATE - n DAYS`), and if the column name, which can be an expression (for example, `DATE(START_TIMESTAMP)`), is less than 18 characters.

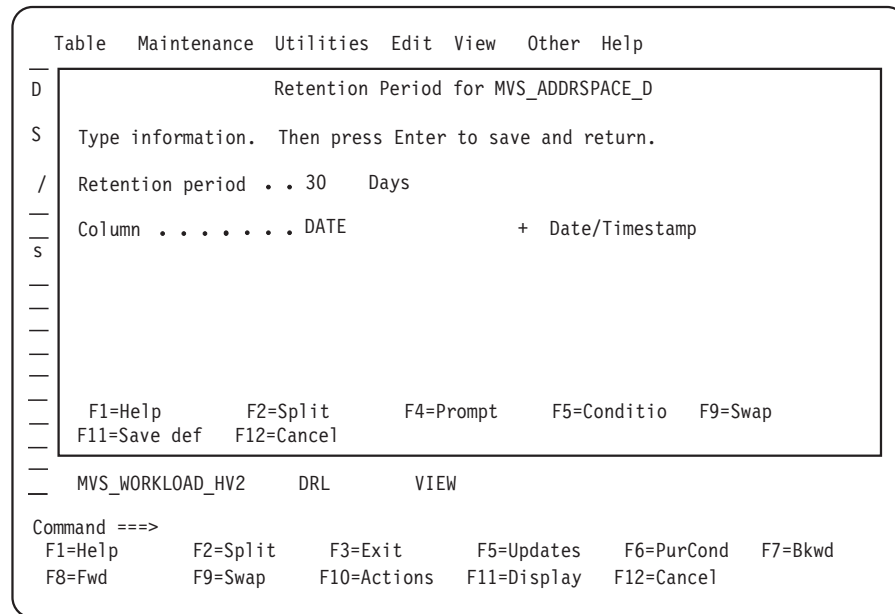


Figure 84. Retention Period window

2. Modify information in the fields. The column is the date or timestamp column in the table that Tivoli Decision Support for z/OS uses to determine the age of the rows.
3. Press Enter.

The changes are saved and you are returned to the Tables window.
4. If the purge condition does not have the standard format, the Purge Condition window is displayed (Figure 85 on page 192) instead of the Retention Period window.

This window is displayed also if you press F5 from the Retention Period window.

Working with tables and update definitions

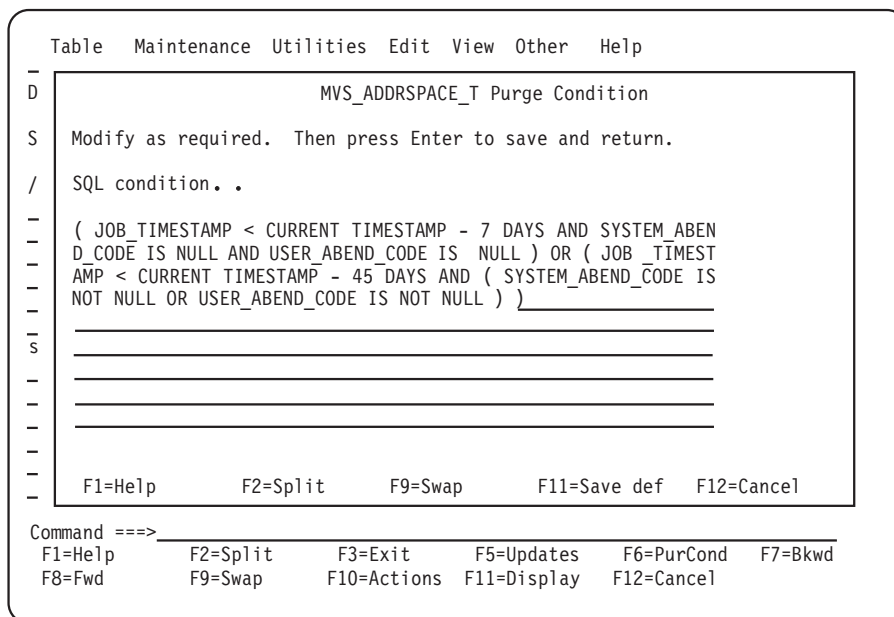


Figure 85. Purge Condition window

5. Modify the SQL condition, and press Enter.
The changes are saved and the previous window is displayed.

Displaying and modifying a table or index space

About this task

Each table in the product database is in a table space, and each index is in an index space. Tivoli Decision Support for z/OS usually uses one table space for each component. You can use the administration dialog to work with table and index spaces.

Note: The method described here makes changes directly to your DB2 database, and does not affect the component definition. You lose such direct changes if you delete and reinstall a component. To change the component definition to keep it in line with the database, use the **Space** pull-down in the **Components** window, in addition to making the direct change as described in the following steps.

To make a change to a table space:

Procedure

1. From the **Tables window**, select the **Maintenance** pull-down. Do not select a table first.
2. The pull-down has these options:
 1. Tablespace...
 2. Index and indexspace...

To change table space parameters, select 1. The **Tablespace** window is displayed (with the Tablespace pull-down illustrating the options available: you can use the Utilities pull-down to reorganize or get statistics on a table space).

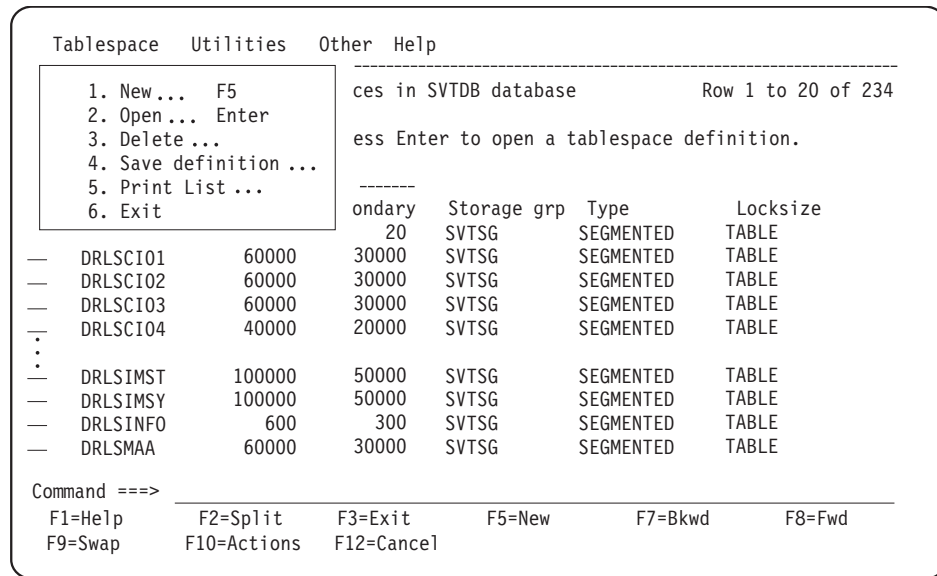


Figure 86. Tablespaces window

You can use the **Save definition** option to create SQL commands that can recreate the selected table space. Note that this does not update the component definition: only the definition of the selected table space is saved.

3. Select a table space and press Enter. The **Tablespace** window is displayed, which you can use to change the table space parameters. Change the parameters and press Enter.

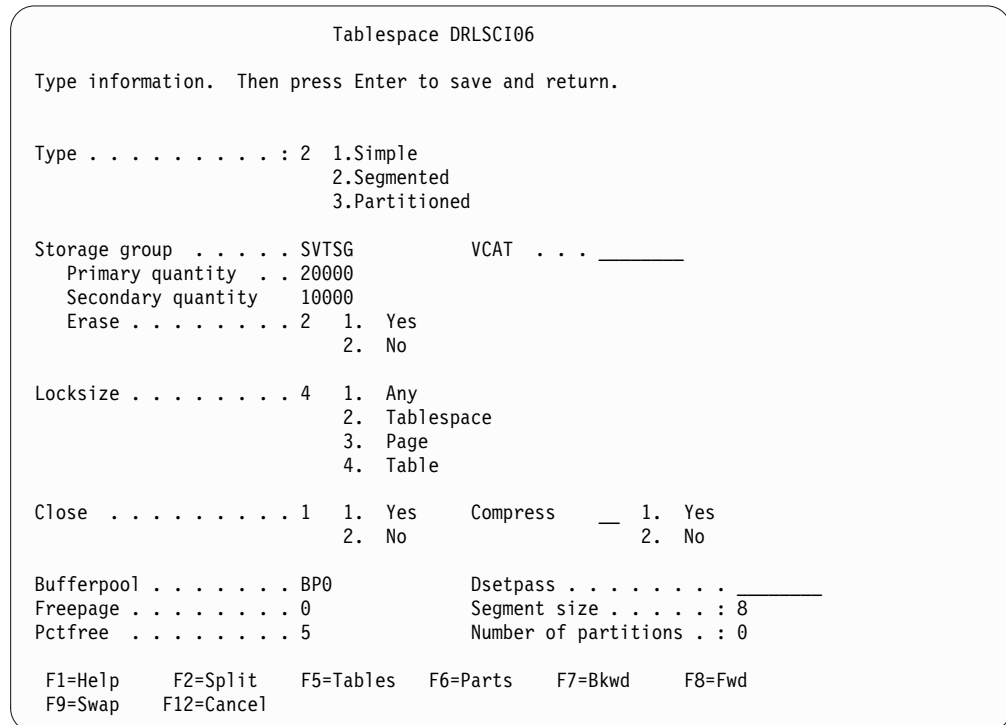


Figure 87. Tablespace DRLxxx

Tivoli Decision Support for z/OS takes action depending on the parameters to be changed:

Working with tables and update definitions

Where reorganization is needed

Some parameter changes need a database reorganization before they take effect. In this case the product :

- a. Makes the change, using the ALTER TABLESPACE command.
- b. Creates a batch job to reorganize the database, which you can submit when it is convenient.

Where the database needs to be stopped

Some parameter changes need exclusive use of the database. In this case the product creates a batch job that:

- a. Stops the database.
- b. Makes the change, using the ALTER TABLESPACE command.
- c. Starts the database again.

Do not submit the job if some task, for example a collect, is using the table space, because this stops the collect job.

In other cases

Some parameter changes can be made immediately. Tivoli Decision Support for z/OS issues the ALTER TABLESPACE command online.

Press F1 to get more information about a parameter, or refer to the discussion of designing a database in DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference.

Making changes to an index space

About this task

To make a change to an index space:

Procedure

1. From the **Tables** window (Figure 63 on page 164), select the **Maintenance** pull-down. Do not select a table first.
2. To change index space parameters, select 2. The **Indexes** window is displayed (with the Index pull-down illustrating the options available; you can use the Utilities pull-down to reorganize an index space).

Table name	Indexspace	Unique	Cluster
AVAILD_IX	AVAILDRI	YES	YES
AVAILM_IX	AVAILMRI	YES	YES
AVAILP_IX	AVAILPRI	YES	YES
AVAILT_IX	AVAILTRI	YES	YES
AVAILW_IX	AVAILWRI	YES	YES
⋮			
CICS_DLI_TRAN_D	CICS1TA1	YES	YES
CICS_DLI_TRAN_W	CICS1TR4	YES	YES
CICS_DLI_USR_D	CICS1SKV	YES	YES

Figure 88. Indexes window

3. Select an index space and press Enter. An index window will be displayed, which you can use to change the index space parameters: change the parameters and press Enter.

Tivoli Decision Support for z/OS takes action depending on the parameters to

```

Index CICS_A_DLI_USR_W
Press Enter to save and return.

Table name . . . . . : CICS_A_DLI_USR_W
Indexspace name . . . . : CICS1S60

Storage group . . . . . SVTSG          VCAT . . . . . _____
Primary quantity . . 20000          Unique . . . . . 1  1. Yes
Secondary quantity  10000          Cluster . . . . . 1  1. Yes
Erase . . . . . 2  1. Yes          Part . . . . . 2  1. Yes
                               2. No          2. No
Close . . . . . 1  1. Yes          2. No

Subpages . . . . . 4              Pctfree . . . . . 10
Bufferpool . . . . . BP0          Dsetpass . . . . . _____
Freepage . . . . . 0

F1=Help   F2=Split   F5=Columns  F6=Parts   F7=Bkwd   F8=Fwd
F9=Swap   F12=Cancel
    
```

Figure 89. Index window

be changed:

Where the index must be recreated

In this case the product:

- Asks you to confirm the change.
- Deletes the index, with the DROP command.
- Redefines the index, using the DEFINE command.

Where the database needs to be stopped

Some parameter changes need exclusive use of the database. In this case the product creates a batch job that:

- Stops the database.
- Makes the change, using the ALTER command.
- Starts the database again.

Do not submit the job if some task, for example a collect, is using the index space, because this stops the collect job.

In other cases

Some parameter changes can be made immediately. Tivoli Decision Support for z/OS issues the ALTER command online.

Press F1 to get more information about a parameter, or refer to the discussion of designing a database in DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference.

Making table space parameter changes that do not require offline or batch action

About this task

If you want just to make table space parameter changes that do not require offline or batch action, you can use this alternative method:

Working with tables and update definitions

Procedure

1. From the **Tables window** (Figure 63 on page 164), select a table in the table space to open.
2. Select 5, Open Tablespace, from the Table pull-down.
Tivoli Decision Support for z/OS displays the **Tablespace window**.

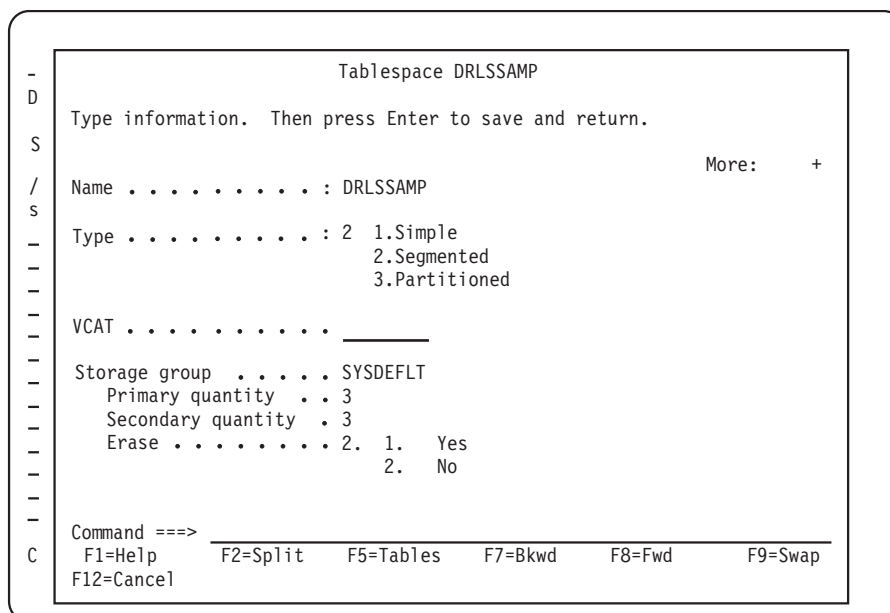


Figure 90. Tablespace window

3. Type any changes in the fields.
Note: You can scroll the window to display more options.
4. Press F5 to see a list of tables in the tablespace.
The **Tables window** is displayed.
5. Press Enter when you finish viewing this window.
You are returned to the **Tablespace window**.
6. Press Enter.
The changes to the tablespace are saved and you are returned to the **Tables window**.

Displaying a view definition

About this task

You can use the administration dialog to display a view definition created with SQL statements.

To display the view definition:

Procedure

1. From the Tables window, select a view to display, and press Enter.
The View window is displayed (Figure 91 on page 197).

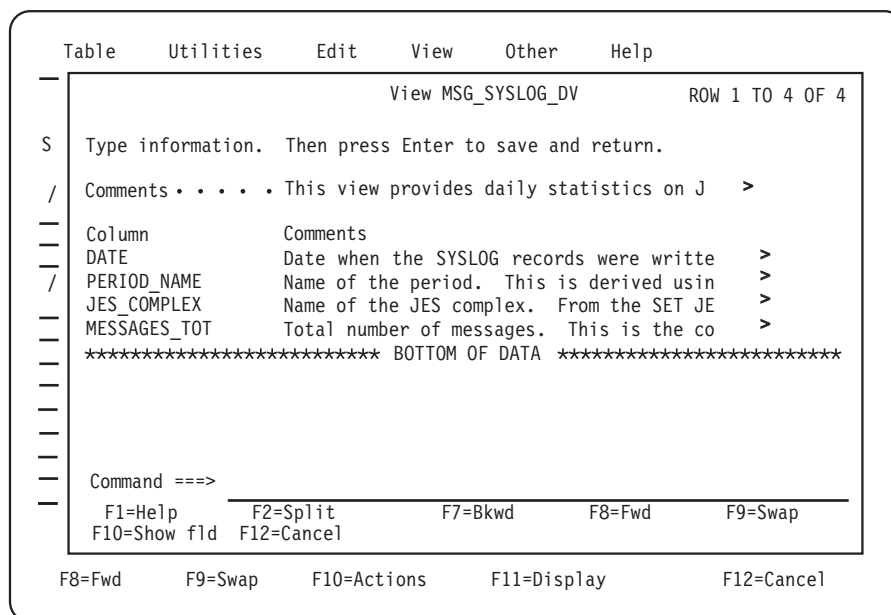


Figure 91. View window

2. You can change any of the comments in a view definition. To change a comment, type the text in the Comments field.
3. Press Enter when you finish displaying the view definition.
The changes are saved and you are returned to the Tables window.

Printing a list of Tivoli Decision Support for z/OS tables

About this task

Tivoli Decision Support for z/OS maintains a list of all tables in the product database. You can use the administration dialog to print a list of these tables.

Procedure

1. From the Table pull-down in the Tables window (Figure 63 on page 164), select 8, Print list.
The Print Options window is displayed.
2. Type the required information, and press Enter.
The list of Tivoli Decision Support for z/OS tables is routed to the destination you specified.

Saving a table definition in a data set

About this task

Each table in the Tivoli Decision Support for z/OS database is defined using SQL. You can use the administration dialog to save the SQL table definition statement in a data set.

To save a table definition statement in a data set:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table definition to save in a data set.

Working with tables and update definitions

2. Select 7, Save definition, from the Table pull-down.
The Save Data Set window is displayed.
3. Type the data set name in the field, and press Enter.
The table definition in the data set that you specified is saved and you are returned to the Tables window.

Listing a subset of tables in the Tables window

About this task

When you select 4, Tables, from the Administration window, all tables in the Tivoli Decision Support for z/OS database are listed in the Tables window. You can use the administration dialog to list only a subset of tables in the Tivoli Decision Support for z/OS database in the Tables window.

To specify which tables should appear in the Tables window:

Procedure

1. From the View pull-down in the Tables window (Figure 63 on page 164), select 2, Some, and press Enter.
Tivoli Decision Support for z/OS displays the Select Table window.
2. Type selection criteria in the fields, and press Enter.

Note: You can see a list of components by pressing F4.

The tables that correspond to the criteria you specified are listed.

To list all the tables, from the View pull-down in the Tables window, select 1, All. All the tables in the Tivoli Decision Support for z/OS database are listed.

Creating a table

About this task

Tivoli Decision Support for z/OS stores data collected from logs in DB2 tables. Each component includes table definitions for tables that it uses. However, you might need to create additional tables.

You can use the administration dialog to create a table. You should have a working knowledge of DB2 databases before attempting to create a table. Refer to the DB2 documentation for more information.

Note: Views cannot be created from the administration dialog. Refer to the DB2 documentation for a description of how to create views using SQL.

To create a table:

Procedure

1. From the Table pull-down in the Tables window (Figure 63 on page 164), select 1, New, and press Enter.
The New Table window is displayed (Figure 92 on page 199).
2. Type required information in the fields.
3. To see a list of available table spaces, place the cursor in the Tablespace field, and press F4.

The Prompt for Tablespace window is displayed. If the table is related to existing tables, you might want to put the table in the same tablespace.

4. Select a table space from the list, and press Enter.

The product returns to the New Table window, and the table space appears in the Tablespace field.

Note: To create a table space, see “Creating a table space” on page 200.

New Table

Type information. Then press F5 to add columns. To select an already added column, press Enter.

Table name . . . _____ Prefix DRL
 Database . . . DRLDB Tablespace . . . _____ +
 Comments . . . _____ >

/ Column	Type	Length	Nulls	Primary Key
***** BOTTOM OF DATA *****				

Command ===> _____

F1=Help	F2=Split	F3=Exit	F4=Prompt	F5=Add col	F6=Indexes
F7=Bkwd	F8=Fwd	F9=Swap	F10=Show fld	F11=Delete	F12=Cancel

Figure 92. New Table window

5. Press F5 to add a column to the table.

The product displays the Add Column window (Figure 75 on page 182).

6. Type the required information in the fields, and press Enter.

You are returned to the Add Column window.

7. When you finish adding columns to the table, press F12.

You are returned to the New Table window.

8. Press F6 to add indexes to the table.

The Indexes window is displayed (Figure 76 on page 183).

9. Press F5 to add an index.

The Add Index window is displayed (Figure 78 on page 184).

10. Type the required information in the fields, and press Enter.

The index is added and you are returned to the Indexes window.

11. Press F3 to return to the New Table window.

12. Press F3 when you finish typing information.

The table is added to the database and you are returned to the Tables window.

To create a table by using an existing table as a template:

Creating a table using an existing table as a template

About this task

You can also create a table by using an existing table as a template.

Working with tables and update definitions

Procedure

1. From the Tables window, select the table to use as a template.
2. Select 1, New, from the Table pull-down.
The New Table window is displayed.

Note: The fields are filled with information from the template table.

3. The rest of the procedure is the same as when creating a table without a template.

Note: The index for the template table is not copied and must be added for the primary key. To add an index, see “Displaying and adding a table index” on page 182.

Using the administration dialog to delete a column from a table:

About this task

You can use the administration dialog to delete a column from a table you are defining. To delete a column:

Procedure

1. From the New Table window, select an existing column.
2. Press F11 to delete the column.
A confirmation window is displayed.
3. Verify the deletion by pressing Enter.
The column is deleted and you are returned to the New Table window.

Deleting a table or view

About this task

To delete a table or view:

Procedure

1. Select the table or view to delete in the Tables window (Figure 63 on page 164) and select 6, Delete, from the Table pull-down.

Note: Tivoli Decision Support for z/OS prevents you from deleting table definitions that affect, or are affected by, other product objects. To delete a table definition, remove links from the table to other product objects.

A confirmation window is displayed.

2. Verify the deletion by pressing Enter.
The table or view is deleted and you are returned to the Tables window.

Note: A table in a partitioned tablespace cannot be explicitly deleted (dropped). You can drop the tablespace that contains it. This does not have any impact on other tables because only one table can be defined in a single tablespace.

Creating a table space

About this task

DB2 tables are in table spaces. For a new table, you might need to create a table space.

You can use the administration dialog to create a table space. You must have some knowledge of DB2 databases before creating the table space. See “Understanding table spaces” on page 96 for more information about table spaces, or refer to the discussion of designing a database in *DB2 Universal Database for OS/390 and z/OS: Administration Guide and Reference*.

To create a tablespace:

Procedure

1. From the New Table window (Figure 92 on page 199), place the cursor in the Tablespace field and press F4.
The Prompt for Tablespace window is displayed.
2. From the Prompt for Tablespace window, press F5.
The New Tablespace window is displayed.
3. Type required information in the fields, and press Enter.
A table space is created and you are returned to the Prompt for Tablespace window.
4. Press Enter again to return to the New Table window.
5. Continue creating the table as described in “Creating a table” on page 198.

Note: It is also possible to create a tablespace without creating a table: use the Maintenance pull-down in the Tables window (as described in “Displaying and modifying a table or index space” on page 192) and select New from the Tablespace pull-down in the Tablespaces window.

Creating an update definition

About this task

In Tivoli Decision Support for z/OS, update definitions specify how to store data from log records in DB2 tables and how to use data from one table to update another. Each component includes all the update definitions that it uses. However, if you tailor the objects used during a collect, or create components of your own, you might need to create more update definitions.

You can use the administration dialog to create an update definition. You can also use log collector language. Refer to the *Language Guide and Reference* for more information about defining update definitions using log collector language.

To create an update definition:

Procedure

1. From the Tables window (Figure 63 on page 164), select a table for addition of an update definition, and press F5.
The Update Definitions window is displayed (Figure 79 on page 185).
2. To use an existing update definition as a template, select one of the update definitions from the list and press F5. Otherwise, do not select an update definition.
The New Update Definition window is displayed. The columns are filled with values from the template.
3. To create an update definition without a template, press F5 from the Update Definitions window.

Working with tables and update definitions

You are prompted for the name of the target table in the Target Table of New Update window. Type the name of the target table, and press Enter.

The New Update Definition window is displayed.

4. Type required information in the fields, and press F3.

The new update definition is saved and you are returned to the Update Definitions window.

You might choose to use abbreviations for expressions in the expression fields. Or you might require that data be distributed over some interval or used in availability processing. See these topics in ““Displaying and modifying update definitions of a table” on page 184” for information:

- “Working with abbreviations” on page 187
- “Modifying a distribution clause” on page 188
- “Modifying an apply schedule clause” on page 189

5. Press F3 again to return to the Tables window.

Deleting an update definition

About this task

Update definitions are supplied for all data tables. You can use the administration dialog to delete an update definition you no longer need. Tivoli Decision Support for z/OS removes all references to the update from its system tables. However, it does not delete the definition member; you can use the dialog to reinstall it.

To delete an update definition of a table:

Procedure

1. From the Tables window (Figure 63 on page 164), select the table and press F5.
The Update Definitions window for the table is displayed (Figure 79 on page 185). All update definitions where the selected table is either the source or the target are included.
2. Select the update definition to delete, and press F11.
A confirmation window is displayed.
3. Verify the deletion by pressing Enter.
The definition is updated and you are returned to the Update Definitions window.
4. Press F3 to return to the Tables window.

Administering user access to tables

About this task

When you install a component, Tivoli Decision Support for z/OS grants read access to the users or groups you have specified in dialog parameters (the default is the DRLUSER group). You can use the administration dialog to grant or revoke table access to other Tivoli Decision Support for z/OS users.

To grant table access to other users:

Procedure

1. From the **Tables** window (Figure 63 on page 164), select one or more tables to grant access to.
2. Select 5, Grant, from the Utilities pull-down.

The **Grant Privilege** window is displayed (Figure 93).

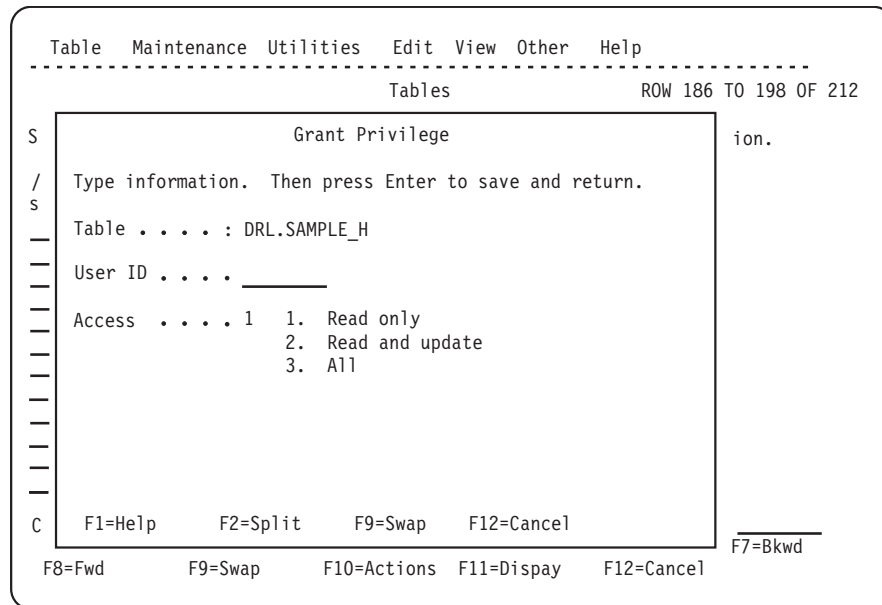


Figure 93. Grant Privilege window

3. Type required information in the fields, and press Enter.
The user ID is granted access to the table.
4. When you finish granting access to the table, press F12.
If you selected more than one table, the **Grant Privilege** window for the next table is displayed. When you complete the **Grant Privilege** window for the last table, you are returned to the **Tables** window.

Revoking table access

About this task

To revoke table access:

Procedure

1. From the **Tables** window (Figure 63 on page 164), select one or more tables to revoke access to.
2. Select 6, Revoke, from the Utilities pull-down.
The **Revoke Privilege** window (Figure 94 on page 204) is displayed.

Working with tables and update definitions

```
Revoke Privilege                                ROW 1 TO 5 OF 5

Select one more user IDs.  Then press Enter to execute.

 / User ID   Table                               Privilege  Grantor
- DRL       DRL.SAMPLE_H                         DELETE     DRL
- DRL       DRL.SAMPLE_H                         UPDATE     DRL
- DRL       DRL.SAMPLE_H                         INSERT     DRL
- DRL       DRL.SAMPLE_H                         SELECT     DRL
- DRLUSER   DRL.SAMPLE_H                         SELECT     DRL

***** BOTTOM OF DATA *****

Command ==>
F1=Help   F2=Split   F7=Bkwd   F8=Fwd    F9=Swap   F12=Cancel
```

Figure 94. Revoke Privilege window

3. Select the user IDs with table access privileges to revoke, and press Enter.
The access privileges are revoked and you are returned to the Tables window.

Chapter 10. Working with the log data manager option

This section contains information about the Tivoli Decision Support for z/OS log data manager option, which automates and simplifies the collection of data.

After providing a summary of the log data manager, this section then describes:

- How the log data manager is invoked from the administration dialog (page “Summary of how the log data manager is used”).
- The job for recording of log data sets for collection (page “Job step for recording a log data set for collection” on page 206).
- Modifying log collector statements to be used in the collect (page “Modifying log collector statements” on page 209).
- Modifying the list of log data sets to be collected (page “Listing and modifying the list of log data sets to be collected” on page 212).
- The collect job and the parameters it uses (page “The DRLJLDMC collect job and the parameters it uses” on page 215).
- Modifying the list of successfully collected log data sets (page “Listing and modifying the list of log data sets to be collected” on page 212).
- Modifying the list of unsuccessfully collected log data sets (page “Modifying the list of unsuccessfully collected log data sets” on page 221).

Summary of how the log data manager is used

You usually include a log data set for use with the log data manager by inserting a job step DRLELDML in the job that creates the log data set. The job step DRLELDML records the log data set as being ready to be collected by the log data manager collect job. You must run the job step DRLELDML for each log data set that you want to be collected.

The log data manager collect job DRLELDMC then performs the data collection and updates the database tables.

You can also use the Administration dialog windows to do the following:

- Amend the list of log data sets to be collected.
- Amend the list of the log data sets that were successfully or unsuccessfully collected.
- Amend the collect statements used in a collect.

Invoking the log data manager

About this task

To invoke the log data manager:

Procedure

1. From the Administration Dialog window, select 3, Logs, to display the Logs window.
2. Select one of the displayed logs, then select 5, Open Log Data Manager (a new option provided with the log data manager) from the Log pull-down. The log

Invoking the log data managerlog data manager

data manager Main Selection window (Figure 95) is displayed.

```
DRLJLDML          Log Data Management of SMF logs
Select one of the following.  Then press Enter.

   1. Log collector statements
   2. Log data set to be collected
   3. Log data sets collected successfully
   4. Log data sets collected with failure

F1=Help  F2=Split  F9=Swap  F12=Cancel
```

Figure 95. Log Data Manager Main Selection window.

Each of these options is discussed in the sections of this chapter.

3. The Main Selection window gives you the possibility to:
 - Browse, add, delete and modify log collector statements.
 - Add, delete, and change the list of log data sets to be collected by the collect job.
 - List the log data sets that were collected successfully by the collect job.
 - List the log data sets that were collected unsuccessfully by the collect job.

Job step for recording a log data set for collection

The job step DRLJLDML records a log data set as being ready to be collected. The collect job DRLJLDMC then performs the collection of this log data set (described in “The DRLJLDMC collect job and the parameters it uses” on page 215).

After job step DRLJLDML has successfully run, Tivoli Decision Support for z/OS will have created a record in system table DRLLDM_LOGDATASETS (described in “DRLLDM_LOGDATASETS” on page 227). You must run this job step for each log data set that you want to be collected by the log data manager. The list of log data sets to be collected can then be displayed, changed, or deleted, or a log data set added for collection (an alternative to using the DRLJLDML job), using the Log Data Sets To Be Collected window, described in “Listing and modifying the list of log data sets to be collected” on page 212.

Using the DRLJLDML job step

About this task

To use the DRLJLDML job step:

Procedure

1. Ensure that your log data sets are cataloged (otherwise the DRLJLDML job step does not work).
2. Take a copy of the supplied sample DRLJLDML job step.
3. Insert the DRLJLDML job step in each job that creates a log data set, and which you want to be collected by the log data manager. For Generation Data Sets, you must insert the DRLJLDML job step after each Generation Data Set member that has been created.
4. Enter the name of the log data set (*.stepname.ddname) in the DRLLLOG DD statement of the job step (described in 207).
5. Run the job you have now amended, to create the log data set.

DRLJLDML sample job

This job is shipped with Tivoli Decision Support for z/OS as sample job DRLJLDML.

DRLJLDML job

```
//DRLJLDML JOB (ACCT#),'LOGS'                                00010014
//*****                                                    00020000
//*                                                         * 00030000
//* Licensed Materials - Property of IBM                    * 00040000
//*                                                         * 00050000
//* 5698-B06 Copyright IBM Corporation 1995, 2015         * 00060000
//* See Copyright instructions.                             * 00070000
//*                                                         * 00080000
//*****                                                    00090000
//*                                                         * 00100000
//* Name: DRLJLDML                                         * 00110000
//*                                                         * 00120000
//* Status: Tivoli Decision Support for zOS 1.8.2          * 00130011
//*                                                         * 00140000
//* Function:                                              * 00150000
//*   Log Data Manager - register a log data set sample job * 00160014
//*                                                         * 00170000
//*   This job is used to register the log data set (only one) * 00180011
//*   specified in DRLLLOG in the DRLLDM_LOGDATASETS as being ready * 00190000
//*   for collect by the Log Data Manager.                 * 00191000
//*                                                         * 00192000
//* Input:                                                 * 00200000
//*   The exec DRLELDML accepts the following parameters:   * 00280000
//*                                                         * 00290000
//* SYSPREFIX=xxxxxxx Prefix for system tables. default=DRLSYS * 00290100
//* PLAN=xxxxxxx DB2 plan name default=DRLPLAN * 00290200
//* SYSTEM=xxxxxx DB2 subsystem name. default=DSN * 00290300
//* SHOWSQL=xxx Show SQL. YES/NO default=NO * 00290400
//* LOGTYPE=xxxxxxxxx Log type (e.g. SMF). Required. * 00290500
//* LOGID=xxxxxx Log ID. If not specified (or '='), a blank * 00290611
//* Log ID is generated, and the default collect * 00290711
//* statement is used in collect. * 00290811
//* ONTAPE=N/Y Specify if the LOG name is on DASD or not. If *
//* not coded, it defaults to NO. *
//*                                                         * 00290900
//* DRLLLOG DD card: Name of log data set to be registered * 00291011
//* (can refer to a previous step). * 00291100
//* It must be cataloged. * 00291200
//*                                                         * 00291300
//* Output: Log data set name registered in * 00291411
//* sysprefix.DRLLDM_LOGDATASETS together * 00291500
//* with LOG_NAME, LOG_ID and TIME_ADDED. * 00291600
//* Confirmation message including data set name * 00292000
//*                                                         * 00650000
//* Notes: * 00660000
//* Before you submit the job, do the following: * 00670000
//* 1. Fill in a correct log data set name. * 00680012
//* 2. Check that the steplib db2loadlibrary is correct. * 00681012
//* 3. Change the input parameters to DRLELDML as required. * 00700013
//* 4. Change the DB2 load library name according to * 00270000
//* the naming convention of your installation. * 00260000
//* Default is 'db2loadlibrary'. * 00260000
//* 5. Change the TDSz high level qualifier. Default id 'DRLvrn'. * 00270000
//*                                                         * 00720000
//* CHANGE ACTIVITY: * 00730000
//* 00 1995-03-05 IW Created * 00740000
//* 01 1997-05-14 GL Added ONTAPE parameter to allow PQ06678 *
//* the use of tape log names PQ06678 *
//*                                                         * 00770000
//*                                                         *
```

Job step for recording a log data set for collection

```
/* CHANGE ACTIVITY: *
/* CHANGE FLAG TYPE DATE DESCRIPTION *
/* -----*
/* $D0=DCR066, TDS181,01/06/09,ADL(SM): Update TDS Version and *
/* DB2 dataset names. *
/* $D1=DCR116, TDS182,15/05/15,ADL(SM): Update TDS Version *
/* *
/****** 00780000
//LDMLLOG EXEC PGM=IKJEFT01 00790000
/* 00800000
//SYSPROC DD DISP=SHR,DSN=DRLVrm.SDRLEXEC 00800111
//STEPLIB DD DISP=SHR,DSN=DRLVrm.SDRLOAD 00800211
// DD DISP=SHR,DSN=db2loadlibrary <-- 00810010
/****** 00870010
/* MESSAGES 00880000
/* 00890000
//DRLOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80) 00900009
/****** 00910010
/* LOG DATA SET 00920000
/* DSN=*.stepname.ddname can be used 00930000
/* 00930100
//DRLLLOG DD DISP=SHR,DSN=... <-- 00931010
/****** 00932010
/* START EXEC DRLELDML 01210002
/* 01220000
//SYSPRINT DD SYSOUT=* 01230000
//SYSPRINT DD SYSOUT=* 01240000
//SYSTSIN DD * 01250000
%DRLELDML SYSTEM=DSN SYSPREFIX=DRLSYS - 01260007
LOGTYPE=SMF - 01270000
LOGID=' ' ONTAPE=N
/* 01340000
```

Setting the parameters for job DRLJLDML

About this task

These are the rules for entering parameter values:

1. LOGTYPE is the only parameter that *must* be changed by you. The remaining parameters are optionally changed by you.
2. Blanks must not exist before or after an equal (=) sign.
3. Blanks must not exist within a parameter value.
4. A parameter value must not be enclosed in apostrophes.
5. A continuation mark (-) can be placed in any column.

The following are the DRLJLDML job parameters:

SYSPREFIX

The prefix of all Tivoli Decision Support for z/OS system and control DB2 tables. If you do not specify a value here, the default DRLSYS is used.

SYSTEM

The DB2 subsystem. The default value is DSN.

PLAN

The name of the DB2 application plan. The default value is DRLPLAN.

SHOWSQL

When this value is set to YES, all executed SQL statements will be written to an output file. The default value is NO.

LOGTYPE, LOGID

Each combination of LOGTYPE and LOGID identifies the collect statements to be used by the collect job (which is run after this job):

- If you do not enter a value for LOGID, or if you enter two apostrophes with no blank between ('), the default collect statements for this LOGTYPE will be used for collecting the log data set.
- If you set LOGID to a user-defined value, the collect statements for the user-defined value will be used for this LOGTYPE, when collecting the log data set.
- Using different values of LOGID will produce more than one collect for a specific LOGTYPE. These collects will normally be run serially. However, you can run these collects in parallel by setting up your system accordingly.

Modifying log collector statements

In order to modify log collector statements, this section describes the following:

- “Listing the data sets containing collect statements”
- “Editing the collect statements”
- “Adding a log ID and collect statements data set” on page 211
- “Changing the collect statements data set name” on page 211

Listing the data sets containing collect statements

About this task

To list the log collector statements used with a log type:

Procedure

Select 1, Log collector statements, from the log data manager Main Selection window. The Collect Statements window (Figure 96) is displayed, one row for each log ID defined for the log type. When a default row is created during installation of a product component, the field log ID is always blank.

```
DRLDLMS      Log Data Manager Collect Statements for SMF

Select a Log ID. Then press Enter to edit the collect statement

/  Log ID      Collect statement data set
s                                DRLxxx.SDRLDEFS (DRLBSMF)
-  MVSA        DRLxxx.LOCAL.DEFS (MVSACOLL)
-  MVSB        DRLxxx.LOCAL.DEFS (MVSBCOLL)
-  MVSX        DRLxxx.LOCAL.DEFS (MVSXCOLL)
-  MVS1        DRLxxx.LOCAL.DEFS (MVS1COLL)
-  SYS1        DRLxxx.LOCAL.DEFS (SYS1COLL)

Command ==>
F1=Help   F2=Split  F3=Exit   F5=Add    F6=Modify F7=Bkwd   F8=Fwd
F9=Swap   F11=Delete F12=Cancel
```

Figure 96. Collect Statements window

Editing the collect statements

About this task

To edit (default action) the collect statements for a log ID:

Modifying log collector statements

Procedure

1. Select the log ID whose collect statements you want to edit, and press Enter. The Edit window (Figure 97) is displayed.
2. Edit the collect statements using the ISPF editor. If the member does not exist, it will be automatically created by the edit. If the collect statements data set does not exist or is not cataloged, an error message is displayed. A confirmation window is displayed if a member of the product definition library is selected for editing. If you want to edit collect statements that reside in the product distribution library, follow the instructions given in “Modifying Tivoli Decision Support for z/OS-supplied collect statements”
3. On completion of the editing, you are returned to the Log Data Manager Collect Statements window.

Results

Note: The COMMIT AFTER BUFFER FULL ONLY parameter will not be accepted in the collect statement member if the collect involves concatenated log data sets (an appropriate error message is displayed). The reason is that such concatenated data sets are never recorded in the DRLLOGDATASETS system table as being collected.

```
EDIT ---- DRLxxx.SDRLDEFS(DRLBSMF)----- COLUMNS 0
***** ***** TOP OF DATA *****
000001 COLLECT SMF;
***** ***** BOTTOM OF DATA *****

COMMAND ==>                                SCROLL ==
F1=Help      F2=Split    F3=Exit     F5=Rfind    F6=Rchange  F7=Up
F8=Down      F9=Swap      F10=Left   F11=Right   F12=Cancel
```

Figure 97. Edit collect statements window

Modifying Tivoli Decision Support for z/OS-supplied collect statements

About this task

Not all the components have a default collect statement supplied by the product. You must modify the collect statements for these log types to use with these components. You might also want to modify other Tivoli Decision Support for z/OS-supplied collect statements. In all cases, a warning is displayed if you attempt to edit a collect statement member that resides in the product distribution library.

Note: Any modifications you make to Tivoli Decision Support for z/OS-supplied collect statements are lost if a PTF or new release updates the member containing the collect statements.

To modify a Tivoli Decision Support for z/OS-supplied collect statement member:

Procedure

1. Copy the member containing the collect statements to your local library.
2. Use option F6=Modify of the Log Data Manager Collect Statements window to change the data set name of the default log ID (see “Modifying log collector statements” on page 209 for details).
3. Edit the collect statements member as you require.

Adding a log ID and collect statements data set

About this task

To add a log ID and data set name to the list:

Procedure

1. Press F5 and the Add Collect Statements Definition window is displayed (Figure 98)
2. Type a log ID and data set name and press Enter. The log ID and data set name are added to the Log Data Manager Collect Statements list in alphanumeric sequence. However, a non-existent data set is not created

```
DRLDLDMA      Add Collect Statements Definition for SMF

Type information. Then press Enter to save.

Log ID _____ (blank for default collect statements)
Data set name _____

F1=Help      F2=Split    F9=Swap     F12=Cancel
```

Figure 98. Add Collect Statements Definition window

Changing the collect statements data set name

About this task

To change the name of a collect statements data set:

Procedure

1. Select the log ID corresponding to the data set name which you want to modify, and press F6. The Modify Collect Statements Definition window is displayed (Figure 99 on page 212)
2. Type the modified data set name and press Enter. The data set name is changed in the Log Data Manager Collect Statements list.

Modifying log collector statements

```
DRLDLDMB          Modify Collect Statements Definition for SMF

Type information.  Then press Enter to save.

Log ID      M_VSA_____
Data set    DRLxxx.LOCAL.DEFS(MVSACOLL)_____

F1=Help     F2=Split    F9=Swap     F12=Cancel
```

Figure 99. Modify Collect Statements Definition window

Listing and modifying the list of log data sets to be collected

In order to list and modify the list of log data sets to be collected, this section describes the following:

- “Listing the log data sets to be collected”
- “Modifying the log ID for a log data set” on page 213
- “Deleting information about a log data set” on page 214
- “Recording a log data set to be collected again” on page 214
- “Adding a log data set to be collected” on page 214

Listing the log data sets to be collected

About this task

To list the log data sets to be collected:

Procedure

Select 2, Log data sets to be collected, from the log data manager Main Selection window. The Log Data Sets To Be Collected window (Figure 100 on page 213) is displayed, one row for each log ID and log data set.

What to do next

Each list of log data sets are sorted firstly by log ID, and then by the date the log data set was added.

Each log data set displayed in this window has a value in the Status column, which can contain one of these values:

- blank

The log data set is ready to be collected by the DRLMLDMC job (see “The DRLJLDMC collect job and the parameters it uses” on page 215 for details).

- 'SELECT'

This value occurs when the log data set has been selected for collect by the DRLMLDMC job, but the collect has not completed. The data set is protected from a collect by a “parallel” invocation of the DRLMLDMC job. If the DRLMLDMC job abends, the action you take depends upon how many log data sets have the status 'SELECT' after the abend has occurred:

Modifying log collector statements

- If there are many log data sets with status 'SELECT', run job DRLELDMC with parameter CLEANUP=YES, to record the log data sets as ready for collection again.
- If there are only a few log data sets with status 'SELECT', it is easier to manually record the data sets as ready for collection again by selecting F4=Rerun for these log data sets.
- A log collector return code or a system or user abend code
This occurs when the log data set was collected with a failure, and the Rerun option was selected for this log data set in the Log Data Sets Collected with Failure window (described in “Modifying the list of unsuccessfully collected log data sets” on page 221). The data set is collected again the next time job DRLELDMC is run.

```
DRLDLDMT      SMF Log Data Sets To Be Collected

Select a data set. Then press Enter to modify Log ID.

/  Log ID      Log data set              Time added      Status
-  MVS        SYS170.SMFLOG.SLOG9501222  2004-11-22.13
s  MVS        SYS170.SMFLOGA.SLOG950122  2004-11-21.23  SELECT
-  MVS        SYS170.SMFLOGB.SLOG950122  2004-11-22.01
-  MVS        SYS170.SMFLOGX.SLOG950122  2004-11-22.01
-  MVS1       SYS170.SMFLOG1.SLOG02      2004-11-21.23      8
-  MVS2       SYS170.SMFLOG.MVS2.SLOG01  2004-11-21.10     U0005
-  SYS1       SYS170.SMFLOG.SYS1.SLOG01  2004-11-18.10      20

Command ==> _____
F1=Help  F2=Split  F3=Exit  F4=Rerun  F5=Add   F7=Bkwd  F8=Fwd
F9=Swap  F11=Delete F12=Cancel
```

Figure 100. SMF Log Data Sets To Be Collected window

Modifying the log ID for a log data set

About this task

To modify the log ID (the default action) to be used with a log data set:

Procedure

1. Select the log ID and press Enter. The Modify Log ID for a Log Data Set window is displayed (Figure 101 on page 214).
2. Type the modified log ID and press Enter. The log ID is then changed in the Log Data Sets To Be Collected list.

Note: You can also use this window to display the full length of a truncated log data set name. Data set names longer than 34 characters are truncated in the Log Data Sets To Be Collected window, but are displayed in full in the Modify Log ID for a Log Data Set window.

Modifying log collector statements

```
DRLDMM    Modify Log ID for a SMF Log Data Set

Type Log ID.  Then press Enter to save.

Log ID    MVSA    (blank for default collect statements)
Data set  SYS150.SMFLOGA.SLOG950122

F1=Help   F2=Split   F9=Swap   F12=Cancel
```

Figure 101. Modify Log ID For a Log Data Set window.

Deleting information about a log data set

About this task

To delete an entry from the Log Data Sets To Be Collected window:

Procedure

1. Select the log ID and log data set and press F11.
2. Press Enter to confirm deletion.

Recording a log data set to be collected again

About this task

A log data set can be recorded for collection again if it has the value 'SELECT' in the Status column, caused by the collect job abending and as a result, the log data set still having the value 'SELECT' in the Status column.

After the log data set has been recorded for collection again, it is included in the next collect job (described in “The DRLJLDMC collect job and the parameters it uses” on page 215).

To record a log data set to be collected again:

Procedure

1. Select the log ID and log data set and press F4
2. Press Enter to confirm.

Adding a log data set to be collected

About this task

To add an entry to the Log Data Sets To Be Collected list:

Procedure

1. Press F5 and the Add a Data Set To Be Collected window is displayed (Figure 102 on page 215).
2. Type the log ID and log data set name and press Enter. The Log Data Sets To Be Collected window is displayed, containing the added entry.

```

DRLDLMN   Add a SMF Data Set To Be Collected

Type information.  Then press Enter to save.

Log ID      _____ (blank for default collect statements)
Data set name _____

F1=Help    F2=Split    F9=Swap    F12=Cancel
    
```

Figure 102. Add a Data Set To Be Collected window

An error message is displayed in this window if you attempt to add an already existing log data set.

The DRLJLDMC collect job and the parameters it uses

The job DRLJLDMC is used to collect log data sets that are recorded as being ready for collection. A system table (described in “DRLDLM_COLLECTSTMT” on page 226) is used to identify the data set containing the collect statements to be used for the collect.

Log data sets are recorded as ready for collection either by running the job DRLJLDMC (see “Job step for recording a log data set for collection” on page 206 for details), or by using the Log Data Sets To Be Collected window (see “Listing and modifying the list of log data sets to be collected” on page 212 for details).

Deciding which log data sets to collect

Using the two parameters LOGTYPE and LOGID you specify which log data sets you want to collect. If you omit both parameters, all log data sets that are ready to be collected are collected. If, however, you decide to enter values for LOGTYPE and LOGID, a subset only of the log data sets belonging to the specified log type is collected.

Concatenation of log data sets

Each time you run the DRLELDMC EXEC, all log data sets corresponding to the values you enter for the parameters LOGTYPE and LOGID are serially collected. The log collector function is used only once for all log data sets of the same log type and log ID. Log data sets are added to the log collector file DRLLOG in the order in which they were recorded by the Log Data Manager. As a result, the log collector output files DRLOUT and DRLDUMP may contain the output from many log data sets.

You should also note that if the collect of such a concatenated log data set fails after one or more log data sets have been successfully collected, the remaining log data sets in the concatenation are not collected. You must then rerun the DRLJLDMC collect job, to collect these remaining log data sets.

Running collect jobs in parallel

If you do not specify the LOGID and/or the LOGTYPE parameters, the DRLELDMC EXEC calls the log collector and runs the collect job each time a combination of log type and log ID is processed. If you want to decrease the total elapsed time of these collects, you can run DRLJLDMC collect jobs in parallel. However, you should not run jobs with the same LOGTYPE parameter in parallel.

The collect job and the parameters it uses

DRLELDMC sample job

This job is shipped with the product as sample job DRLJLDMC.

DRLJLDMC

```
//DRLJLDMC JOB (ACCT#),'COLLECT'                                00010000
//*****                                                    00020000
//*                                                            * 00030000
//* Licensed Materials - Property of IBM                      * 00040000
//*                                                            * 00050000
//* 5698-B06 Copyright IBM Corporation 2003, 2015            * 00060001
//* See Copyright instructions.                               * 00070000
//*                                                            * 00080000
//*****                                                    00090000
//*                                                            * 00100000
//* Name: DRLJLDMC                                           * 00110000
//*                                                            * 00120000
//* Status: Tivoli Decision Support for zOS 1.8.2            * 00130001
//*                                                            * 00140000
//* Function:                                                 * 00150000
//*   Log Data Manager Collect Log Data Sets sample job     * 00160000
//*                                                            * 00170000
//*   This job is used to collect log data sets that are recorded * 00180000
//*   in the DRLLDLM_LOGDATASETS system table as being ready for * 00190000
//*   collect by the Log Data Manager.                       * 00191000
//*                                                            * 00192000
//* Input:                                                    * 00193000
//*   The exec DRLELDMC accepts the following parameters:    * 00194000
//*                                                            * 00195000
//* SYSPREFIX=xxxxxxx Prefix for system tables. default=DRLSYS * 00196000
//* SYSTEM=xxxxxxx   DB2 subsystem name. default=DSN          * 00197000
//* PREFIX=xxxxxxx   Prefix for all other tables.default=DRL  * 00198000
//* PLAN=xxxxxxx     DB2 plan name default=DRLPLAN            * 00198100
//* DSPREFIX=xxxxxxx Prefix for creation of data sets DRLOUT and * 00198200
//*                                                            DRLDUMP. default=DRL * 00198300
//* SHOWSQL=xxx      Show SQL. YES/NO default=NO             * 00198400
//* SHOWINPUT=xxx    Copy DRLIN to DRLOUT. YES/NO default=YES * 00199000
//* LOGTYPE=xxxxxxxxx Log type (e.g. SMF). If not specified, * 00199100
//*                                                            all log types are selected for processing. * 00199200
//* LOGID=xxxxxxx    Log ID. If not specified, all log id's are * 00199300
//*                                                            are selected for processing. Default Log ID * 00199400
//*                                                            should be coded as '='. * 00199500
//* RETENTION=xxx    Retention period for DRLOUT, DRLDUMP and * 00200000
//*                                                            collect result info. default=10 days * 00210000
//* PURGE=xxx        Purge info for successful collects that * 00220000
//*                                                            are older than its Retention period * 00230000
//*                                                            YES/NO default=YES * 00240000
//* CLEANUP=xxx      Option only to be used after an Abend. * 00240100
//*                                                            No collect is done. Processes only log data * 00240200
//*                                                            sets marked with SELECT in the Log Data Sets * 00240300
//*                                                            To Be Collected list (on panel DRLDLDMT). * 00240400
//*                                                            Output: the data set being collected when * 00240500
//*                                                            the abend occurred will be moved to the * 00240600
//*                                                            Collected With Failure list. Other concate- * 00240700
//*                                                            nated data sets are moved to the Successful * 00240800
//*                                                            list or made ready for a renewed collect. * 00240900
//*                                                            YES/NO default=NO * 00241000
//*                                                            * 00242000
//* DRLOUT/DRLDUMP DD card: if any of these files are specified * 00250000
//*                                                            they will be used by all collects started * 00260000
//*                                                            by this job. They will then not be controlled * 00261000
//*                                                            or viewed by the Log Data Manager dialog. * 00270000
//*                                                            * 00270100
//* DRLLOG DD card: Must not be allocated. * 00270200
//*                                                            * 00270300
//* LMDLOG EXEC card: The value used for DYNAMNBR should be * 00270400
```


The collect job and the parameters it uses

```

/*          as a minimum, 2 plus the number of          * 00270500
/*          log data sets to be collected.              * 00270600
/*          * 00280000
/* Output:      The results of the collects are recorded in * 00290000
/*          sysprefix.DRLLDM_LOGDATASETS together      * 00291000
/*          with LOG_NAME, LOG_ID and TIME_ADDED.      * 00291100
/*          Job messages in the DRLMSG file            * 00291200
/*          * 00291300
/* Notes:       * 00291400
/*   Before you submit the job, do the following:     * 00291500
/*   1. Check that the steplib db2loadlibrary is correct. * 00291800
/*   2. Change the parameters to DRLELDMC as required. * 00292000
/*   3. Change the DB2 load library name according to * 00292101
/*       the naming convention of your installation.   * 00292201
/*       Default is 'db2loadlibrary'.                 * 00292301
/*   4. Change the TDSz data set HLQ (default is DRLvrn.) * 00292401
/*          * 00292500
/* CHANGE ACTIVITY: * 00292600
/*   00 1995-03-05 IW Created                          * 00292700
/*          * 00293000
/*          * 00293101
/* CHANGE ACTIVITY: * 00293201
/* CHANGE FLAG TYPE DATE DESCRIPTION                  * 00293301
/* -----* 00293401
/* $D0=DCR066, TDS180,01/06/07,ADL(SM): Update TDS Version and * 00293501
/*          DB2 dataset names.                          * 00293601
/* $D1=DCR116, TDS181,15/05/09,ADL(RC): Update TDS Version * 00293701
/*          * 00293801
/* ***** 00294000
//LDMLLOG EXEC PGM=IKJEFT01,DYNAMNBR=20              * 00295000
/*          * 00296000
//SYSPROC DD DISP=SHR,DSN=DRLvrn.SDRLEXEC           -- * 00297001
//STEPLIB DD DISP=SHR,DSN=DRLvrn.SDRLOAD            -- * 00298001
//          DD DISP=SHR,DSN=db2loadlibrary           -- * 00299000
/* ***** 00300100
/**DRLOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)         * 00301000
/**DRLDUMP DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)        * 00302000
/* ***** 00303000
/* MESSAGES                                           * 00320000
/*          * 00330000
//DRLMSG DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)         * 00331000
/* ***** 00331400
/* Add the next three DD statements if you collect IMS. * 00331600
/* Note 1: IMSVER must specify the same release as the * 00331700
/* collect statement used by the Log Data Manager.     * 00331800
/* Note 2: DRLICHKI must be DUMMY or point out an empty * 00331900
/* data set after an IMS restart.                      * 00332000
/* ***** 00332100
/**DRLICHKI DD DSN=Generation data set(0),DISP=SHR   * 00332200
/**DRLICHKO DD DSN=Generation data set(+1),DISP(NEW,CATLG) * 00332300
/**DRLIPARM DD *                                       * 00332400
/**IMSVER=71 -- IMS release being processed. 71 is default * 00332600
/**MAXOUTPUT=50 -- Allow up to 50 outputs per transaction/BMP * 00332800
/**MAXUOR=50 -- Allow up to 50 UOR's per BMP           * 00332900
/* ***** 00333000
/* START EXEC DRLELDMC                                * 00410000
/*          * 00430000
//SYSPRINT DD SYSOUT=*                                * 00440000
//SYSPRT DD SYSOUT=*                                  * 00450000
//SYSTSIN DD *                                        * 00460000
%DRLELDMC SYSTEM=DSN SYSPREFIX=DRLSYS PREFIX=DRL - * 00470000
  DSPREFIX=DRL - * 00480000
  LOGTYPE=SMF - * 00481000
  LOGID='' * 00490000
/*          * 00510000

```

The collect job and the parameters it uses

Setting the DYNAMNBR value

The value for the EXEC parameter DYNAMNBR should be a minimum of the number of log data sets to be collected, plus 2.

The supplied default is 20.

Setting the parameters for job DRLJLDMC

The rules for entering parameter values are as follows:

1. All parameters are *optional*.
2. Blanks must not exist before or after an equal sign (=).
3. Blanks must not exist within a parameter value.
4. A parameter value must not be enclosed in apostrophes.
5. A continuation mark (-) can be placed in any column.

These are the DRLJLDMC job parameters:

SYSPREFIX

The prefix of all product system and control DB2 tables. If you do not specify a value here, the default DRLSYS is used.

SYSTEM

The DB2 subsystem. The default value is DSN.

PREFIX

The prefix used with all other tables. The default value is DRL.

PLAN

The name of the DB2 application plan. The default value is DRLPLAN.

DSPREFIX

The prefix used for the creation of data sets DRLOUT and DRLDUMP. The default is DRL. The names of these data sets are 'dsprefix_value.Ddate.Ttime.DRLOUT/DRLDUMP' where *date* and *time* are generated. The maximum length of DSPREFIX is 20 characters.

SHOWSQL

When this value is set to YES, all executed SQL statements are written to an output file. The default value is NO.

SHOWINPUT

When this value is set to YES, all DRLIN statements are written to DRLOUT. The default value is YES.

LOGTYPE, LOGID

Each combination of LOGTYPE and LOGID identifies the log IDs to be used in the collect. If log type is not specified, all log types are selected for processing. If log ID is not specified, all log IDs for the log type specified are selected for processing. The default log ID is selected by setting this value to straight quotes (").

RETENTION

The retention period for DRLOUT, DRLDUMP and the log data manager information that is produced by the collects. The default is 10 days.

PURGE

This parameter determines whether or not the information resulting from successful collects should be purged when the date of the information is older than the retention period. The parameter can be set to the value YES or NO. If

The collect job and the parameters it uses

PURGE is set to YES, all log data manager information about successfully collected log data sets is deleted (for all log types and log IDs). The default value is PURGE=YES.

CLEANUP

This parameter is used when the DRLELDMC job has had an abend during a collect of *concatenated* log data sets. If you run the DRLELDMC job with parameter CLEANUP set to YES, log data sets that were successfully collected before the abend occurred are moved to the Log Data Sets Successfully Collected list. The log data set that was being collected *when* the abend occurred is moved to the Log Data Sets Collected With Failure list. The default value is CLEANUP=NO.

DRLOUT DD statement

If this file is specified, it is used by all collects started by this job. However, this file is not used by the log data manager dialog.

DRLDUMP DD statement

If this file is specified, it is used by all collects started by this job. However, this file is not used by the log data manager dialog.

DRLLLOG DD statement

Must not be allocated.

Modifying the list of successfully collected log data sets

About this task

To list the log data sets that have been successfully collected:

Procedure

Select 3, Log data sets collected successfully, from the log data manager Main Selection window.

The Log Data Sets Collected Successfully window (Figure 103) is displayed, one row for each log data set that has been successfully collected by the Log Data Manager for this log type.

The list of data sets are sorted by the Time collected column.

```
DRLDLDMC      Log Data Sets Collected Successfully for SMF

Select a data set. Then press Enter to view DRLOUT.

/  Log data set          Time collected      RC
-  SYS170.SMFLOGX.SLOG950120  2004-11-21.02.03.25  0
-  SYS170.SMFLOGB.SLOG950120  2004-11-21.01.33.25  0
-  SYS170.SMFLOGA.SLOG950120  2004-11-21.01.15.10  0
-  SYS170.SMFLOG.SLOG950120B  2004-11-21.01.01.20  0
-  SYS170.SMFLOG.SLOG950120A  2004-11-21.00.45.20  0
-  SYS170.SMFLOGA.SLOG950119  2004-11-20.23.15.10  0
-  SYS170.SMFLOG.SLOG950119B  2004-11-20.01.45.20  0
-  SYS170.SMFLOGB.SLOG950119  2004-11-20.01.13.25  0
-  SYS170.SMFLOGX.SLOG950119  2004-11-20.01.13.25  0
-  SYS170.SMFLOG.SLOG950119A  2004-11-20.00.45.20  0

Command ==>>
F1=Help      F2=Split      F3=Exit      F5=DRLDUMP    F6=Retent.    F7=Bkwd
F8=Fwd       F9=Swap       F11=Delete   F12=Cancel
```

Figure 103. Log Data Sets Collected Successfully window

Modifying the list of successfully collected log data sets

Viewing the information about successfully collected log data sets

About this task

To view the log data manager information about a log data set (the default action):

Procedure

Select a log data set and press Enter. The DRLOUT data set is displayed in ISPF Browse mode (if a DRLOUT statement was not included in the collect job).

Viewing the dump data set

Procedure

Select the log data set and press F5. The DRLDUMP data set is displayed using the ISPF Browse function, if a DRLDUMP DD statement was not present in the collect job. DRLDUMP should be empty if the return code from the collect was 0.

Changing the retention period of information about a log data set

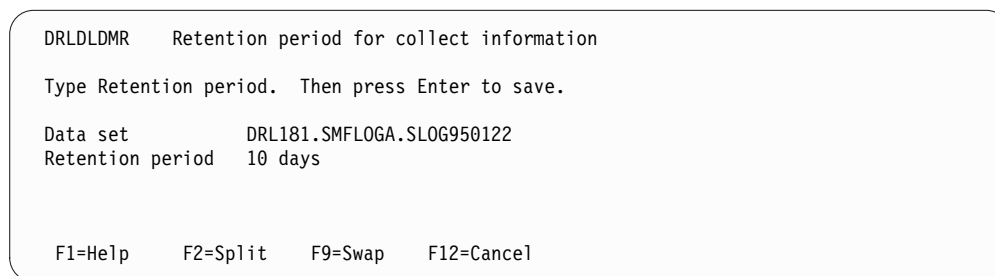
About this task

To change the retention period for the log data manager log data manager information about a log data set:

Procedure

1. Select the log data set and press F6. The Retention Period for Collect Information window is displayed (Figure 104).
2. Type the retention period field the number of days you require, and press Enter

Note: You are not changing the retention period for the collected log data here, but only the retention period for the log data manager information about the log data set.



```
DRLDLDMR  Retention period for collect information

Type Retention period.  Then press Enter to save.

Data set      DRL181.SMFLOGA.SLOG950122
Retention period  10 days

F1=Help      F2=Split    F9=Swap     F12=Cancel
```

Figure 104. Retention Period window

Deleting the information about a log data set

About this task

To delete the log data manager information about a log data set together with DRLOUT and DRLDUMP data sets (if they exist):

Modifying the list of successfully collected log data sets

Procedure

1. Select the log data set for which you want to delete the log data manager information from, and press F11.
2. Press Enter to confirm deletion.

Note: You are not deleting the log data set itself, but only the log data manager information about the log data set.

Modifying the list of unsuccessfully collected log data sets

About this task

To list the log data sets that have been unsuccessfully collected:

Procedure

Select 4, Log Data Sets Collected with Failure, from the log data manager Main Selection window. The Log Data Sets Collected with Failure window (Figure 105) is displayed, one row for each log data set that has been unsuccessfully collected by the Log Data Manager for this log type.

The list of data sets are sorted by the Time collected column.

```
DRDLDMF      Log Data Sets Collected with Failure for SMF
Select a data set. Then press Enter to view DRLOUT.

/  Log data set           Time collected      RC
_  SYS170.SMFLOG1.SLOG01  2004-11-20.23.22.10  8
_  SYS170.SMFLOG.SYS1.SLOG0 2004-11-18.10.16.22 20

Command ==>>>
F1=Help      F2=Split      F3=Exit      F4=Rerun      F5=DRLDUMP    F7=Bkwd
F8=Fwd       F9=Swap       F11=Delete   F12=Cancel
```

Figure 105. Log Data Sets Collected with Failure window

Viewing the unsuccessfully collected log data set

About this task

To view the log data set (the default action):

Procedure

1. Select the log data set and press Enter.
2. The DRLOUT data set is displayed in ISPF Browse mode (if a DRLOUT statement was not included in the collect job).

Viewing the dump data set

About this task

To view the dump data set (DRLDUMP):

Procedure

Select the log data set and press F5. The DRLDUMP data set is displayed using the ISPF Browse function, if a DRLDUMP DD statement was not present in the collect

Modifying the list of successfully collected log data sets

job. DRLDUMP is empty in most cases if the return code from the collect was 0.

Recording a log data set to be collected again

About this task

If you record a log data set for collection again, it is included in the next collect job (described in “The DRLJLDMC collect job and the parameters it uses” on page 215).

However, the entry you select to be collected again is not deleted from the Log Data Sets Collected with Failure window.

If you select a log data set to be collected a second time (using the F4=Rerun option) after it has already been successfully collected, the log collector detects this incorrect selection and the collect attempt is rejected. However, if you have specified REPROCESS=YES in the collect job to recollect a successfully collected log data set, the log collector does not reject the collect.

To record a log data set to be collected again:

Procedure

1. Select the log data set.
2. Press F4. An error message is displayed if this log data set is already included in the list of data sets to be collected.

Deleting the information about a log data set

About this task

To delete the information about a log data set from the list shown, together with DRLOUT and DRLDUMP data sets (if they exist):

Procedure

1. Select the log data set you want to delete, and press F11.
2. Press Enter to confirm deletion.

Part 4. Administration reference

Chapter 11. System tables and views

This section describes system tables and views. These tables are used by the product log collector and dialogs. They are created during installation of the product base, with the prefix for system tables specified in *userid.DRLFPROF*. The default prefix for the tables is DRLSYS.

System tables do not appear in the tables list in the administration dialog.

Each table description includes information about the table, a description of each key column and data column in the table, and an example of the table's contents.

Key columns are marked with a "K".

Data columns are listed after the last key column.

The tables appear in alphabetic order, with any underscores ignored.

Log collector system tables

These tables contain definitions used by the log collector. They are maintained by the log collector. **Do not** modify them.

DRLEXPRESSIONS

This system table contains one row for each expression or condition in a log, record, record procedure, or update definition.

Column name		Data type	Description
OBJECT_TYPE	K	CHAR(8)	Object type. This is LOG, RECORD, RECPROC, or UPDATE.
OBJECT_NAME	K	VARCHAR(18)	Name of the object.
EXPRESSION_NO	K	SMALLINT	Expression sequence number within the object.
EXPRESSION		VARCHAR(2000)	Original expression text.
PARSED_EXPRESSION		VARCHAR(2000)	Parsed version of the expression.

DRLFIELDS

This system table contains one row for every field in each defined record type.

Column name		Data type	Description
RECORD_NAME	K	VARCHAR(18)	Name of the record. For a log header, this is *log-name*.
FIELD_NO	K	SMALLINT	Field sequence number within the record.
FIELD_NAME		VARCHAR(18)	Name of the field.

Log collector system tables

Column name		Data type	Description																																																										
TYPE		CHAR(8)	Type of the field. The following values are possible: <table border="0"> <thead> <tr> <th>Type</th> <th>Field Format</th> </tr> </thead> <tbody> <tr><td>BINARY</td><td>BINARY</td></tr> <tr><td>BINARYS</td><td>BINARY SIGNED</td></tr> <tr><td>EINTEGER</td><td>EXTERNAL INTEGER</td></tr> <tr><td>HEXIN</td><td>EXTERNAL HEX</td></tr> <tr><td>DECIMAL</td><td>DECIMAL(p,s)</td></tr> <tr><td>ZONED</td><td>ZONED(p,s)</td></tr> <tr><td>FLOAT</td><td>FLOAT</td></tr> <tr><td>CHAR</td><td>CHAR or CHAR(n)</td></tr> <tr><td>CHAR(*)</td><td>CHAR(*) or LENGTH * CHAR</td></tr> <tr><td>VARCHAR</td><td>VARCHAR</td></tr> <tr><td>BIT</td><td>BIT or BIT(n)</td></tr> <tr><td>HEX</td><td>HEX</td></tr> <tr><td>DATE_001</td><td>DATE(0CYDDDF)</td></tr> <tr><td>DATE_002</td><td>DATE(YYYYDDDF)</td></tr> <tr><td>DATE_003</td><td>DATE(MMDDYY)</td></tr> <tr><td>DATE_004</td><td>DATE(YYDDDF)</td></tr> <tr><td>DATE_005</td><td>DATE(CYYMMDDF)</td></tr> <tr><td>DATE_006</td><td>DATE(YMMDD)</td></tr> <tr><td>DATE_007</td><td>DATE(MMDDYYYY)</td></tr> <tr><td>TIME_001</td><td>TIME(1/100S)</td></tr> <tr><td>TIME_002</td><td>TIME(HHMMSSTF)</td></tr> <tr><td>TIME_003</td><td>TIME(0HHMSSF)</td></tr> <tr><td>TIME_004</td><td>TIME(HHMMSSTH)</td></tr> <tr><td>TIME_005</td><td>TIME(HHMMSSXF)</td></tr> <tr><td>TIME_006</td><td>TIMER(HHMMSS)</td></tr> <tr><td>TIME_007</td><td>TIME(HHMMSSU6)I</td></tr> <tr><td>NTV_001</td><td>INTV(MMSSTTF)</td></tr> <tr><td>TSTAMP_</td><td>1TIMESTAMP(TOD)</td></tr> </tbody> </table>	Type	Field Format	BINARY	BINARY	BINARYS	BINARY SIGNED	EINTEGER	EXTERNAL INTEGER	HEXIN	EXTERNAL HEX	DECIMAL	DECIMAL(p,s)	ZONED	ZONED(p,s)	FLOAT	FLOAT	CHAR	CHAR or CHAR(n)	CHAR(*)	CHAR(*) or LENGTH * CHAR	VARCHAR	VARCHAR	BIT	BIT or BIT(n)	HEX	HEX	DATE_001	DATE(0CYDDDF)	DATE_002	DATE(YYYYDDDF)	DATE_003	DATE(MMDDYY)	DATE_004	DATE(YYDDDF)	DATE_005	DATE(CYYMMDDF)	DATE_006	DATE(YMMDD)	DATE_007	DATE(MMDDYYYY)	TIME_001	TIME(1/100S)	TIME_002	TIME(HHMMSSTF)	TIME_003	TIME(0HHMSSF)	TIME_004	TIME(HHMMSSTH)	TIME_005	TIME(HHMMSSXF)	TIME_006	TIMER(HHMMSS)	TIME_007	TIME(HHMMSSU6)I	NTV_001	INTV(MMSSTTF)	TSTAMP_	1TIMESTAMP(TOD)
Type	Field Format																																																												
BINARY	BINARY																																																												
BINARYS	BINARY SIGNED																																																												
EINTEGER	EXTERNAL INTEGER																																																												
HEXIN	EXTERNAL HEX																																																												
DECIMAL	DECIMAL(p,s)																																																												
ZONED	ZONED(p,s)																																																												
FLOAT	FLOAT																																																												
CHAR	CHAR or CHAR(n)																																																												
CHAR(*)	CHAR(*) or LENGTH * CHAR																																																												
VARCHAR	VARCHAR																																																												
BIT	BIT or BIT(n)																																																												
HEX	HEX																																																												
DATE_001	DATE(0CYDDDF)																																																												
DATE_002	DATE(YYYYDDDF)																																																												
DATE_003	DATE(MMDDYY)																																																												
DATE_004	DATE(YYDDDF)																																																												
DATE_005	DATE(CYYMMDDF)																																																												
DATE_006	DATE(YMMDD)																																																												
DATE_007	DATE(MMDDYYYY)																																																												
TIME_001	TIME(1/100S)																																																												
TIME_002	TIME(HHMMSSTF)																																																												
TIME_003	TIME(0HHMSSF)																																																												
TIME_004	TIME(HHMMSSTH)																																																												
TIME_005	TIME(HHMMSSXF)																																																												
TIME_006	TIMER(HHMMSS)																																																												
TIME_007	TIME(HHMMSSU6)I																																																												
NTV_001	INTV(MMSSTTF)																																																												
TSTAMP_	1TIMESTAMP(TOD)																																																												
LENGTH		SMALLINT	Length of the field. For DECIMAL and ZONED fields, this is a 1-byte precision followed by a 1-byte scale.																																																										
OFFSET		SMALLINT	Offset of the field in the record or section.																																																										
INSECTION_NO		SMALLINT	Number of the section where the field is contained. This is zero if the field is not in a section.																																																										
REMARKS		VARCHAR(254)	Description of the field, set by the COMMENT ON statement.																																																										

DRLLDM_COLLECTSTMT

This system table contains one row for each combination of log type and log ID that is defined to the Log Data Manager. Each row identifies the collect statement that is used for the log type/log ID combination.

Column name		Data type	Description
LOG_NAME	K	VARCHAR(18)	Name of the log type.
LOG_ID	K	CHAR(8)	The log ID.
COLLECT_STMT_DS		VARCHAR(54)	Name of the data set that contain the collect statement, including the member name (for a PDS member).

DRLLDM_LOGDATASETS

This system table contains one or more rows for each log data set recorded by the Log Data Manager.

Column name		Data type	Description
DATASET_NAME	K	VARCHAR(54)	Name of the log data set, including the member name (for a PDS member).
LOG_NAME	K	VARCHAR(18)	Name of the log type.
TIME_COLLECTED	K	TIMESTAMP	Timestamp of the collect. For a data set not yet collected it is 0001-01-01-00.00.000000. For a successfully collected data set it is set to the value of the TIME_COLLECTED field in the corresponding entry in DRLLLOGDATASETS. For an unsuccessfully collected data set, or a successfully collected data set in which no record was recognized, it set to the timestamp when DRLELDMC called the log collector.
LOG_ID		CHAR(8)	The log ID currently associated with this data set.
TIME_ADDED		TIMESTAMP	Timestamp when the log data set was first recorded.
TIME_COLLECT_CALL		TIMESTAMP	Timestamp when the DRLELDMC exec called the log collector to process the log data set.
COLLECT_RC		CHAR(5)	The return code from the collect. It is blank if not yet collected; '0' or '4' if successfully collected; >= '8' if unsuccessfully collected without abend; 'Unn' if the collect ended with a user abend; 'Snn' if the collect ended with a system abend.
OUTPUT_DS		VARCHAR(35)	The high level qualifiers used when DRLOUT and/or DRLDUMP data sets were created. 'OUTPUT_DS_value.DRLOUT' is the data set name of the DRLOUT file. This value is blank if no DRLOUT or DRLDUMP data set has been created.
RETENTION		SMALLINT	Retention period in days. Null field if not yet collected.
RETENTION_DATE		INTEGER	Collect date expressed as number of days from January 1, Year 1. This field is used for purge calculations. Null field if not yet collected.
COMPLETE		CHAR(1)	Flag indicating the status of the log data set. It is blank if the data set is ready to be collected; 'S' if the collect is running; 'Y' if successfully collected; 'F' if collected with failure.

DRLLLOGDATASETS

This system table contains one row for each collected log data set.

Column name		Data type	Description
LOG_NAME	K	VARCHAR(18)	Name of the log definition.
FIRST_RECORD	K	VARCHAR(80)	First 80 bytes of the first identified record in the data set. This is used to identify the data set and make sure that it is not collected again. If the record is a user defined one, avoid beginning the record with data needed to distinguish two records. For more information, refer to <i>Language Guide and Reference</i> , SH19-6817.
DATASET_NAME	K	VARCHAR(54)	Name of the data set, including the member name (for a PDS member).

Log collector system tables

Column name		Data type	Description
COMPLETE		CHAR(1)	Shows whether the data set has been completely processed. This is Y (the data set has been completely processed) or N (the data set has only been partly processed).
ELAPSED_SECONDS		INTEGER	Collect elapsed time, in seconds. The actual collect elapsed time is a bit longer since there is some activity after this table has been updated.
FIRST_TIMESTAMP		TIMESTAMP	Timestamp of the first record in the log. This is only set if TIMESTAMP expression is specified for the log.
LOG_SOURCE		CHAR(16)	Reserved.
LAST_TIMESTAMP		TIMESTAMP	Timestamp of the last record in the log. This is only set if TIMESTAMP expression is specified for the log.
NCOLLECTS		SMALLINT	Number of times the data set has been collected. If this is greater than 1, it means that collect has been run with the REPROCESS operand to collect the data set again.
NRECORDS		INTEGER	Number of records read from the log data set.
NSELECTED		INTEGER	Number of records identified.
NSKIPPED		INTEGER	Number of records skipped due to timestamp overlap (applies when ON TIMESTAMP OVERLAP SKIP specified).
NUPDATES		INTEGER	Number of database rows updated when the data set was collected.
NINSERTS		INTEGER	Number of database rows inserted when the data set was collected.
NDELETES		INTEGER	Number of database rows deleted when the data set was collected.
RETURN_CODE		SMALLINT	Return code from collect; 0 or 4.
TIME_COLLECTED		TIMESTAMP	Date and time when collect ended.
USER_ID		CHAR(8)	ID of the user running collect.
VOLUME		CHAR(6)	Volume serial number for the data set.

DRLLOGS

This system table contains one row for each defined log type.

Column name		Data type	Description
LOG_NAME	K	VARCHAR(18)	Name of the log.
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBMIBM, the value is IBM. <i>mmm</i> [.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
HEADER		CHAR(1)	Shows whether a header is defined for the log. This is Y (a header is defined) or N (no header is defined). If there is a header, it is contained in the DRLRECORDS and DRLFIELDS tables.
TIMESTAMP_EXPR_NO		SMALLINT	Number of the TIMESTAMP expression in the DRLEXPRESSIONS table. This is zero if no TIMESTAMP expression is specified.

Column name		Data type	Description
FIRST_CONDITION_NO		SMALLINT	Number of the FIRST RECORD condition in the DRLEXPRESSIONS table. This is zero if no FIRST RECORD condition is specified.
LAST_CONDITION_NO		SMALLINT	Number of the LAST RECORD condition in the DRLEXPRESSIONS table. This is zero if no LAST RECORD condition is specified.
LOGPROC		CHAR(8)	Name of the log procedure to use for the log. This is blank if no log procedure is specified.
LOGPROC_LANGUAGE		CHAR(8)	Programming language that the log procedure is written in. This is ASM or C.
LOGPROC_PARM_NO		SMALLINT	Number of the log procedure PARM expression in the DRLEXPRESSIONS table. This is zero if no PARM expression is specified.
TIME_DEFINED		TIMESTAMP	Date and time when the log was defined.
CREATOR		CHAR(8)	ID of the user who defined the log.
REMARKS		VARCHAR(254)	Description of the log, set by the COMMENT ON statement.

DRLPURGECOND

This system table contains one row for each purge condition in defined data tables.

Column name		Data type	Description
TABLE_PREFIX	K	CHAR(8)	Prefix of the table.
TABLE_NAME	K	VARCHAR(18)	Name of the table.
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBM, the value is IBM. <i>mmm</i> [.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
SQL_CONDITION		VARCHAR(254)	An SQL condition that defines rows to be deleted from the database when the PURGE statement is executed.
TIME_DEFINED		TIMESTAMP	Date and time when the purge condition was defined.
CREATOR		CHAR(8)	ID of the user who defined the purge condition.

DRLRECORDPROCS

This system table contains one row for each defined record procedure.

Column name		Data type	Description
PROGRAM_NAME	K	CHAR(8)	Name of the record procedure (name of the load module that gets invoked).
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBM, the value is IBM. <i>mmm</i> [.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
LANGUAGE		CHAR(8)	Programming language that the record procedure is written in. This is ASM or C.

Log collector system tables

Column name		Data type	Description
PARAMETER_EXPR_NO		SMALLINT	Number of the PARM expression in the DRLEXPRESSIONS table. This is zero if no PARM expression is specified.
TIME_DEFINED		TIMESTAMP	Date and time when the record procedure was defined.
CREATOR		CHAR(8)	ID of the user who defined the record procedure.
REMARKS		VARCHAR(254)	Description of the record procedure, set by the COMMENT ON statement.

DRLRECORDS

This system table contains one row for each defined record type and one row for each defined header in log definitions.

Column name		Data type	Description
RECORD_NAME	K	VARCHAR(18)	Name of the record. For a log header, this is *log-name*.
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBM, the value is IBM.mmm[.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
LOG_NAME		VARCHAR(18)	Name of the log that contains the record.
BUILT_BY		CHAR(8)	Name of the record procedure that builds the record, if any.
NFIELDS		SMALLINT	Number of fields in the record.
NSECTIONS		SMALLINT	Number of sections in the record.
CONDITION_NO		SMALLINT	Number of the IDENTIFIED BY condition in the DRLEXPRESSIONS table. This is zero if no IDENTIFIED BY condition is specified.
TIME_DEFINED		TIMESTAMP	Date and time when the record was defined.
CREATOR		CHAR(8)	ID of the user who defined the record.
REMARKS		VARCHAR(254)	Description of the record, set by the COMMENT ON statement.

DRLRPROCINPUT

This system table contains one row for every defined record type that must be processed by a record procedure.

Column name		Data type	Description
PROGRAM_NAME	K	CHAR(8)	Name of the record procedure.
RECORD_NAME	K	VARCHAR(18)	Name of the record that is input to the record procedure.

DRLSECTIONS

This system table contains one row for every defined section in defined record types.

Column name		Data type	Description
RECORD_NAME	K	VARCHAR(18)	Name of the record.
SECTION_NO	K	SMALLINT	Section sequence number within the record.
SECTION_NAME		VARCHAR(18)	Name of the section.

Column name		Data type	Description
CONDITION_NO		SMALLINT	Number of the PRESENT IF condition in the DRLEXPRESSIONS table. This is zero if no PRESENT IF condition is specified.
OFFSET_EXPR_NO		SMALLINT	Number of the OFFSET expression in the DRLEXPRESSIONS table. This is zero if no OFFSET expression is specified.
LENGTH_EXPR_NO		SMALLINT	Number of the LENGTH expression in the DRLEXPRESSIONS table. This is zero if no LENGTH expression is specified.
NUMBER_EXPR_NO		SMALLINT	Number of the NUMBER expression in the DRLEXPRESSIONS table. This is zero if no NUMBER expression is specified.
INSECTION_NO		SMALLINT	Number of the section that this section is contained in. This is zero if the section is not contained in another section.
REPEATED		CHAR(1)	Shows whether the section is repeated. This is Y (the section is repeated) or N (the section is not repeated).

DRLUPDATECOLS

This system table contains one row for every column in each update definition, including GROUP BY, SET, and MERGE columns.

Column name		Data type	Description
UPDATE_NAME	K	VARCHAR(18)	Name of the update definition.
UPDATECOL_NO	K	SMALLINT	Sequence number of the column in the update definition.
COLUMN_NAME		VARCHAR(18)	Name of the column.
COLUMN_NO		SMALLINT	Number of the column in the table definition.
FUNCTION		CHAR(8)	This is blank for GROUP BY columns; SUM, MAX, MIN, COUNT, FIRST, LAST, AVG, or PERCENT for SET columns; or INTTYPE, START, END, or QUIET for MERGE columns.
EXPRESSION_NO		SMALLINT	Number of the expression in the DRLEXPRESSIONS table.
COUNT_COLUMN		VARCHAR(18)	If the function is AVG or PERCENT, this contains the name of the column that contains the count of values.
PERCENTILE		SMALLINT	If the function is PERCENT, this contains the percentile value (1 - 99).

DRLUPDATEDISTR

This system table contains one row for every distributed field or column in each update definition.

Column name		Data type	Description
UPDATE_NAME	K	VARCHAR(18)	Name of the update definition.
DISTR_NO	K	SMALLINT	Field or column sequence number in the DISTRIBUTE clause.
FIELD_NAME		VARCHAR(18)	Name of the field or column to be distributed.

DRLUPDATELETS

This system table contains one row for every identifier in the LET clause of each update definition. (The identifiers are defined as *abbreviations* in the administration dialog.)

Log collector system tables

Column name		Data type	Description
UPDATE_NAME	K	VARCHAR(18)	Name of the update definition.
LET_NO	K	SMALLINT	Sequence number of the identifier in the LET clause.
LET_NAME		VARCHAR(18)	Name of the identifier.
EXPRESSION_NO		SMALLINT	Number of the expression in the DRLEXPRESSIONS table.

DRLUPDATES

This system table contains one row for each update definition.

Column name		Data type	Description
UPDATE_NAME	K	VARCHAR(18)	Name of the update definition.
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBMIBM, the value is IBM. <i>mmm</i> [.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
SOURCE_PREFIX		CHAR(8)	Prefix of the source table. This is blank if the source is a record.
SOURCE_NAME		VARCHAR(18)	Name of the source. This is a record name or a table name.
TARGET_PREFIX		CHAR(8)	Prefix of the target table.
TARGET_NAME		VARCHAR(18)	Name of the target table.
SECTION_NAME		VARCHAR(18)	Name of the repeated section, if any, that is used in the update definition.
CONDITION_NO		SMALLINT	Number of the WHERE condition in the DRLEXPRESSIONS table. This is zero if no WHERE condition is specified.
NLETS		SMALLINT	Number of identifiers specified in the LET clause.
NUPDATECOLS		SMALLINT	Number of columns in the GROUP BY, SET, and MERGE clauses.
SCHEDULE_EXPR_NO		SMALLINT	Number of the APPLY SCHEDULE expression in the DRLEXPRESSIONS table. This is zero if APPLY SCHEDULE is not specified.
SCHEDULE_INTTYPE		VARCHAR(18)	Name of the source column or field that defines the interval type.
SCHEDULE_START		VARCHAR(18)	Name of the source column or field that defines the interval start timestamp.
SCHEDULE_END		VARCHAR(18)	Name of the source column or field that defines the interval end time stamp.
SCHEDULE_STATUS		VARCHAR(18)	Name of the identifier that contains the schedule status.
NDISTR_FIELDS		SMALLINT	Number of fields or columns that are distributed.
DISTR_BY_EXPR_NO		SMALLINT	Number of the DISTRIBUTE BY expression in the DRLEXPRESSIONS table. This is zero if DISTRIBUTE is not specified.
DISTR_FROM_EXPR_NO		SMALLINT	Number of the DISTRIBUTE FROM expression in the DRLEXPRESSIONS table. This is zero if DISTRIBUTE is not specified.

Column name		Data type	Description
DISTR_TO_EXPR_NO		SMALLINT	Number of the DISTRIBUTE TO expression in the DRLEXPRESSIONS table. This is zero if DISTRIBUTE is not specified.
DISTR_TIMESTAMP		VARCHAR(18)	Name of the identifier that contains the distribution interval start timestamp.
DISTR_INTERVAL		VARCHAR(18)	Name of the identifier that contains the distribution interval length.
TIME_DEFINED		TIMESTAMP	Date and time when the update was defined.
CREATOR		CHAR(8)	ID of the user who defined the update.
REMARKS		VARCHAR(254)	Description of the update definition, set by the COMMENT ON statement.

GENERATE_PROFILES

This system table contains one row for each GENERATE profile. It is used when installing components that use the GENERATE statement to create the table space, partitioning, and index.

Column name		Data type	Description
PROFILE	K	VARCHAR(18)	Profile name, this value is specified on the PROFILE parameter of the GENERATE TABLESPACE and GENERATE INDEX statements.
COMPONENT_ID	K	VARCHAR(18)	Tivoli Decision Support for z/OS Component ID or %. Allows for a unique profile for a Tivoli Decision Support for z/OS Component..
SUBCOMPONENT_ID	K	VARCHAR(18)	Tivoli Decision Support for z/OS Subcomponent ID or %. Allows for a unique profile for a Tivoli Decision Support for z/OS subcomponent, COMPONENT_ID must be specified before SUBCOMPONENT_ID is valid.
TABLESPACE_NAME	K	VARCHAR(18)	Table space name or %. Allows for a unique profile for a table space. COMPONENT_ID and SUBCOMPONENT_ID are NOT required to use this key field.
TABLESPACE_TYPE		VARCHAR(9)	Table space type (Range, Growth, Segmented). If invalid Growth will be used. For type RANGE there must be a set of definitions in the GENERATE_KEYS system table with the same profile name.
MAXPARTS		INTEGER	Maximum partitions. Used for table space type Growth or Segmented.
NUMPARTS		INTEGER	Initial number of partitions. Used for table space type Growth. (For Range partitioning NUMPARTS is calculated from the number of entries in the GENERATE_KEYS table).
SEGSIZE		INTEGER	Segment size for all table space types.

Log collector system tables

Column name		Data type	Description
TBSPACE1		VARCHAR(250)	The first set of SQL options allowed for GENERATE TABLESPACE. (In syntax diagram for CREATE TABLESPACE in the SQL Reference, TBSPACE1 contains parameters that are in the select group prior to the parameter DSSIZE).
TBSPACE2		VARCHAR(250)	The second set of SQL options allowed for GENERATE DEFAULT TABLESPACE. (In syntax diagram for CREATE TABLESPACE in the SQL Reference, TBSPACE2 contains parameters that are in the select group following the parameter SEGSIZE parameter).
INDEX1		VARCHAR(250)	The first set of SQL options allowed for GENERATE INDEX. (In syntax diagram for CREATE INDEX in the SQL Reference, INDEX1 contains parameters that are in the select group following the parameter 'partition-element').
INDEX2		VARCHAR(250)	The second set of SQL options allowed for GENERATE INDEX. (In syntax diagram for CREATE INDEX in the SQL Reference, INDEX2 contains parameters that are in the select group starting 'BUFFERPOOL-bpname').

GENERATE_KEYS

Column name		Data type	Description
PROFILE	K	VARCHAR(18)	Profile name, this is value specified on the PROFILE parameter of the GENERATE PARTITIONING statement.
COMPONENT_ID	K	VARCHAR(18)	Tivoli Decision Support for z/OS Component ID or %. Allows for a unique profile for a Tivoli Decision Support for z/OS Component.
SUBCOMPONENT_ID	K	VARCHAR(18)	Tivoli Decision Support for z/OS subcomponent ID or %. Allows for a unique profile for a Tivoli Decision Support for z/OS subcomponent, COMPONENT_ID must be specified before SUBCOMPONENT_ID is valid.
TABSPACE_NAME	K	VARCHAR(18)	Table space name or %. Allows for a unique profile for a table space. COMPONENT_ID and SUBCOMPONENT_ID are NOT required to use this key field.
PART_NUM	K	INTEGER	The number of the physical partition for the table space. NUMPARTS for a RANGE table space is the number of PART_NUM entries for this profile name.
PARTITION_OPTIONS		VARCHAR(250)	
RANGE_COLUMNS		VARCHAR(250)	Specifies the columns of the key.
PARTITION_KEY		VARCHAR(250)	The limit key for the partition boundary.

Column name		Data type	Description
INCLUSIVE		CHAR(1)	Specifies if the range values are included in the data partition: Either Y (for Inclusive) or N (for not inclusive).

Dialog system tables

These tables contain definitions used by Tivoli Decision Support for z/OS dialogs and utilities. **Do not** modify them.

DRLCHARTS

This system table stores information extracted from the host graphical report formats (ADMCFORM data). Data is inserted into this table at installation time by the host DRLIRD2 member. If GDDM version 3 or later is installed and available, DRLCHARTS is also updated by the host exec DRLECHRT when a report is saved in the host ISPF dialog.

Column name		Data type	Description
CHART_NAME	K	CHAR(8)	ADMCFORM name. This is the same as the CHART column in the DRLREPORTS table.
TYPE		SMALLINT	This column shows a number identifying the chart type: 1 Line chart 2 Surface chart 3 Histogram 41, 42, 43 Bar chart. The 4 indicates that this is a bar chart; 1, 2, or 3 indicates whether the bars are side by side (1), stacked (2), or overlaid (3). 5 Pie chart 6 Venn diagram 7 Polar chart 8 Tower diagram 9 Table. This is not used. 10 Combination chart.
VALUES		SMALLINT	This column contains one of the values 0, 1, 2, or 3. The column is valid only for chart types 4 (bar) and 5 (pie). For bar charts, the values are: 0 No values are shown 1 Values are shown at the top/end of the bar 2 Values are shown inside the bars 3 Values are shown as in GDDM version 1 release 3 For pie charts, the values are: 1 Values are shown 2 No values are shown
AXIS_ORIENT		SMALLINT	Axis orientation. This can be 1 or 2. 1 means vertical y-axis and bars. 2 means horizontal y-axis and bars.

Dialog system tables

Column name		Data type	Description
Y_DATA_TYPE		VARCHAR(50)	If the chart type is 10 (combination), this column shows the chart type for each data group: 1 Line chart 2 Surface chart 3 Histogram 41, 42, 43 Bar chart For example, 1, 42, 42, 42 identifies a combination chart with a line chart and stacked bars. For a bar chart, the number is concatenated to indicate bar position as in TYPE above.
X_AXIS_TITLE		VARCHAR(52)	This is a string containing the x-axis title.
Y_AXIS_TITLE		VARCHAR(52)	This is a string containing the y-axis title.

DRLCOMPONENTS

This system table contains one row for each Tivoli Decision Support for z/OS component.

Column name		Data type	Description
COMPONENT_NAME	K	VARCHAR(18)	Name of the component.
DESCRIPTION		VARCHAR(50)	Description of the component that is shown in the dialog.
STATUS		CHAR(1)	Component status. This is blank if the component is not installed, I if the component is installed online, or B if the component is installed in batch.
TIME_INSTALLED		TIMESTAMP	Date and time when the component was installed or defined.
USER_ID		CHAR(8)	ID of the user who installed or defined the component.

DRLCOMP_OBJECTS

This system table contains one row for every object in each component.

Column name		Data type	Description
COMPONENT_NAME	K	VARCHAR(18)	Name of the component.
OBJECT_NAME	K	VARCHAR(18)	Name of the object.
OBJECT_TYPE	K	CHAR(8)	Type of object. This is LOG, RECORD, RECPROC, TABSPACE, LOOKUP, TABLE, UPDATE, REPORT, or REPGROUP.
MEMBER_NAME		CHAR(8)	Name of the member in the SDRLEDFS or SDRLRxxx library where the object is defined.
PART_NAME		VARCHAR(18)	Name of the component part that the object belongs to, if any.
EXCLUDE_FLAG		CHAR(1)	Flag to determine if this object is excluded from installation of the component.

DRLCOMP_PARTS

This system table contains one row for every part in each component.

Column name		Data type	Description
COMPONENT_NAME	K	VARCHAR(18)	Name of the component.

Column name		Data type	Description
PART_NAME	K	VARCHAR(18)	Name of the component part.
DESCRIPTION		VARCHAR(50)	Description of the component part that is shown in the dialog.
STATUS		CHAR(1)	Component part status. This is blank if the component part is not installed, I if the component part is installed online, or B if the component is installed in batch.
TIME_INSTALLED		TIMESTAMP	Date and time when the component part was installed or defined.
USER_ID		CHAR(8)	ID of the user who installed or defined the component part.

DRLGROUPS

This system table contains one row for each defined report group.

Column name		Data type	Description
GROUP_NAME	K	VARCHAR(18)	Group ID.
GROUP_OWNER	K	CHAR(8)	Owner of the group. This is blank for a public group.
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBMIBM, the value is IBM. <i>mmm</i> [.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
DESCRIPTION		VARCHAR(50)	Description of the group that is shown in the dialog.
TIME_CREATED		TIMESTAMP	Date and time when the group was defined.
CREATOR		CHAR(8)	ID of the user who defined the group.

DRLGROUP_REPORTS

This system table contains one row for every report in each defined report group.

Column name		Data type	Description
GROUP_NAME	K	VARCHAR(18)	Group ID.
GROUP_OWNER	K	CHAR(8)	Owner of the group.
REPORT_NAME	K	VARCHAR(18)	ID of the report that belongs to the group.
REPORT_OWNER	K	CHAR(8)	Owner of the report that belongs to the group.

DRLREPORTS

This system table contains one row for each defined report.

Column name		Data type	Description
REPORT_NAME	K	VARCHAR(18)	Report ID.
REPORT_OWNER	K	CHAR(8)	Owner of the report. This is blank for a public report.
VERSION		VARCHAR(18)	Version level. The value of VERSION is set for an object when the object is defined and is taken from the value of keyword VERSION. For definitions supplied by IBMIBM, the value is IBM. <i>mmm</i> [.APAR_number], where <i>mmm</i> is the version, release, modification level of the object.
DESCRIPTION		VARCHAR(50)	Description of the report that is shown in the dialog.

Dialog system tables

Column name		Data type	Description
TYPE		CHAR(8)	Type of report. This is QUERY, TABDATA, or GRAPH.
BATCH		CHAR(1)	Y if the report should be produced in batch; N otherwise.
PRINT		CHAR(1)	Y if the report should be printed when produced in batch; N otherwise.
SAVE		CHAR(1)	Y if the report should be saved when produced in batch; N otherwise.
RUN_CYCLE		CHAR(8)	Batch run cycle for the report. This is DAILY, WEEKLY, or MONTHLY.
QUERY_PREFIX		CHAR(8)	Prefix of the QMF query that should be run when the report is produced.
QUERY		VARCHAR(18)	Name of the QMF query that should be run when the report is produced.
FORM_PREFIX		CHAR(8)	Prefix of the QMF form that should be used when the report is produced.
FORM		VARCHAR(18)	Name of the QMF form that should be used when the report is produced.
CHART		CHAR(8)	Name of the GDDM-ICU format to be used for the report. Blank means that the report is tabular.
FILE		CHAR(8)	Name of the member where the data is saved (if type is TABDATA or GRAPH), or where the data should be saved when the report is produced in batch (if save is Y).
MACRO		CHAR(8)	Not used.
TABLE_NAME		VARCHAR(254)	Name of the table or tables on which the the report is bases. This is extracted from the query when the report is defined.
NVARIABLES		SMALLINT	Number of variables defined for the report or extracted from the query.
NATTRIBUTES		SMALLINT	Number of attributes defined for the report.
TIME_CREATED		TIMESTAMP	Date and time when the report was defined.
CREATOR		CHAR(8)	ID of the user who defined the report.
REMARKS		VARCHAR(254)	Long free-format description of the report that can be entered from the dialog.
FINAL_SUMMARY		CHAR(3)	This is valid when QMF is not used. If FINAL_SUMMARY is set to YES, a row containing totals for all numeric columns is generated at the end of the report.
ACROSS_SUMMARY		CHAR(3)	If ACROSS_SUMMARY is set to YES for a report of the Across type, a summary column is created to the right in the report. It contains one total value for each row. This is valid when QMF is not used.

DRLREPORT_ATTR

This system table contains one row for every attribute in each defined report.

Column name		Data type	Description
REPORT_NAME	K	VARCHAR(18)	Report ID.
REPORT_OWNER	K	CHAR(8)	Owner of the report. This is blank for a public report.
ATTRIBUTE_NO	K	SMALLINT	Attribute sequence number.
ATTRIBUTE		VARCHAR(18)	Attribute value.

DRLREPORT_COLUMNS

This system table contains one row for every column in each defined report if QMF is not used. The information is taken from the QMF form.

Column name		Data type	Description
REPORT_NAME	K	VARCHAR(18)	Report ID.
REPORT_OWNER	K	CHAR(8)	Owner of the report. This is blank for a public report.
COLUMN_NO	K	SMALLINT	Column number.
HEADING		VARCHAR(40)	Column heading.
USAGE		CHAR(7)	Usage code.
INDENT		SMALLINT	Column indentation.
WIDTH		SMALLINT	Column width.
EDIT		CHAR(5)	Edit code.
SEQ		SMALLINT	Column sequence number.
DEFINITION		VARCHAR(50)	The DEFINITION column can define an additional report column, which is not present in the SQL query. The definition must be a valid REXX expression, and may contain numeric constants and variables of the <i>&n</i> type, where <i>n</i> is an existing column number. The DEFINITION column is intended only for existing Tivoli Decision Support for z/OS reports and is not used for user-defined reports.

DRLREPORT_QUERIES

This system table contains one row for every query line in each defined report, if QMF is not used.

Column name		Data type	Description
REPORT_NAME	K	VARCHAR(18)	Report ID.
REPORT_OWNER	K	CHAR(8)	Owner of the report. This is blank for a public report.
LINE_NO	K	SMALLINT	Line number in the query.
QUERY_LINE		VARCHAR(80)	Query text.

DRLREPORT_TEXT

This system table is used for host reports when QMF is not used. It contains one row for every heading and footing row. It also contains one row if there is a final summary line with a final text, and one row if there is an expression that limits the number of output rows in the report.

Column name		Data type	Description
REPORT_NAME	K	VARCHAR(18)	Report ID.
REPORT_OWNER	K	CHAR(8)	Owner of the report. This is blank for a public report.
TYPE	K	CHAR(8)	Text type. This is HEADING, FOOTING, DETAIL, FINAL or ROWS.
LINE_NO	K	SMALLINT	Line number for HEADING and FOOTING.
ALIGNMENT		CHAR(6)	Shows how the text should be aligned; left, center, or right.
TEXT		VARCHAR(55)	Text for one line of a report text (see TYPE above).

Dialog system tables

DRLREPORT_VARS

This system table contains one row for every variable in each defined report. The variables may be specified in the DEFINE REPORT statement or extracted from the query.

Column name		Data type	Description
REPORT_NAME	K	VARCHAR(18)	Report ID.
REPORT_OWNER	K	CHAR(8)	Owner of the report. This is blank for a public report.
VARIABLE_NO	K	SMALLINT	Sequence number of the variable.
VARIABLE_NAME		VARCHAR(18)	Name of the variable.
EXPRESSION		VARCHAR(80)	Expression in the query that is compared with the variable, if the variable is found in the query. This is used, with TABLE_NAME in the DRLREPORTS table, to find a list of possible values for the variable.
OPERATOR		CHAR(4)	Operator that is used when comparing the variable and the expression, if the variable is found in the query. This is =, <=, >=, IN, or LIKE.
DATA_TYPE		CHAR(8)	Data type of the variable, if specified. This is CHAR, NUMERIC, DATE, TIME, or TIMESTAMP.
REQUIRED		CHAR(1)	Shows whether the variable must be given a value. This is Y for yes, or N or blank for no.
DEFAULT		VARCHAR(40)	Default value to use for the variable, if specified.
IN_HEADER		CHAR(1)	Variable to determine if the Tivoli Decision Support for z/OS variable is used in the header. This is Y for yes, or N for no.
IN_HEADER_VALUE		VARCHAR(35)	Default header value for a non-required variable without a substitution value.

DRLSEARCH_ATTR

This system table contains one row for every attribute in each saved report search.

Column name		Data type	Description
SEARCH_NAME	K	VARCHAR(18)	Name of the saved search.
SEARCH_OWNER	K	CHAR(8)	Owner of the saved search. This is blank for a public search.
ATTR_SET_NO	K	SMALLINT	Attribute set sequence number. The attribute sets are logically ORed together.
ATTRIBUTE_NO	K	SMALLINT	Attribute sequence number within the attribute set. The attributes within a set are logically ANDed together.
ATTRIBUTE		VARCHAR(18)	Attribute value. This can contain global search characters.

DRLSEARCHES

This system table contains one row for each saved report search.

Column name		Data type	Description
SEARCH_NAME	K	VARCHAR(18)	Name of the saved search.
SEARCH_OWNER	K	CHAR(8)	Owner of the saved search. This is blank for a public search.
DESCRIPTION		VARCHAR(50)	Description of the search that is shown in the dialog.

Column name		Data type	Description
NATTR_SETS		SMALLINT	Number of attribute sets used in the search.
REPORT_DESC		VARCHAR(50)	Report description used in the search. This can contain global search characters.
REPORT_TYPE		CHAR(8)	Report type specified in the search.
REPORT_OWNER		CHAR(8)	Report owner specified in the search.
TIME_CREATED		TIMESTAMP	Date and time when the search was saved.
CREATOR		CHAR(8)	ID of the user who saved the search.

Views on DB2 and QMF tables

These views on DB2 tables are required for users without access to the tables.

View name	Description
DRLCOLUMNS	This view is based on SYSIBM.SYSCOLUMNS in the DB2 catalog. It is used to get column names and comments.
DRLINDEXES	This table is based on SYSIBM.SYSINDEXES in the DB2 catalog. It is used to get table index information.
DRLINDEXPART	This view is based on SYSIBM.SYSINDEXPART in the DB2 catalog. It is used to get index partition information.
DRLKEYS	This view is based on SYSIBM.SYSKEYS in the DB2 catalog. It is used to get information on index keys.
DRLOBJECT_DATA	This view is based on Q.OBJECT_DATA, a QMF control table that contains information about QMF objects.
DRLTABAUTH	This view is based on SYSIBM.SYSTABAUTH in the DB2 catalog. It is used to get table privilege information.
DRLTABLEPART	This view is based on SYSIBM.SYSTABLEPART in the DB2 catalog. It is used to get tablespace information.
DRLTABLES	This view is based on SYSIBM.SYSTABLES in the DB2 catalog. It is used to get a list of tables and comments for the tables.
DRLTABLESPACE	This view is based on SYSIBM.SYSTABLESPACE in the DB2 catalog. It is used to get a list of tablespaces.
DRLVIEWS	This view is based on SYSIBM.SYSVIEWS in the DB2 catalog. It is used to get view definitions.

Views on Tivoli Decision Support for z/OS system tables

These views on Tivoli Decision Support for z/OS dialog system tables are required for users without access to the tables.

View Name	Description
DRLUSER_GROUPPREPS	This view is based on DRLGROUP_REPORTS. It allows a user to update only his own report groups.
DRLUSER_GROUPS	This view is based on DRLGROUPS. It allows a user to update only his own report groups.
DRLUSER_REPORTATTR	This view is based on DRLREPORT_ATTR. It allows a user to update only his own reports.

Views

View Name	Description
DRLUSER_REPORTS	This view is based on DRLREPORTS. It allows a user to update only his own reports.
DRLUSER_REPORTVARS	This view is based on DRLREPORT_VARS. It allows a user to update only his own reports.
DRLUSER_SEARCHATTR	This view is based on DRLSEARCH_ATTR. It allows a user to update only his own searches.
DRLUSER_SEARCHES	This view is based on DRLSEARCHES. It allows a user to update only his own searches.
DRLUSER_REPORTQRY	This view is based on DRLREPORT_QUERIES. It allows a user to update only his own reports.
DRLUSER_REPORTCOLS	This view is based on DRLREPORT_COLUMNS. It allows a user to update only his own reports.
DRLUSER_REPORTTEXT	This view is based on DRLREPORT_TEXT. It allows a user to update only his own reports.

Chapter 12. Control tables and common tables

This chapter describes control tables and common tables. These tables are used by many Tivoli Decision Support for z/OS components. The tables are provided with the Tivoli Decision Support for z/OS base.

Each table description includes information about the table, and a description of each key column and data column in the table.

Key columns are marked with a “K”.

Data columns come after the last key column and are sorted in alphabetic order, with any underscores ignored.

The tables appear in alphabetic order, with any underscores ignored.

Note: Data tables with similar contents (that is, data tables with the same name but different suffixes) are described under one heading. For example, “AVAILABILITY_D, _W, _M” on page 247 contains information about three similar tables:

```
AVAILABILITY_D
AVAILABILITY_W
AVAILABILITY_M
```

Except for the DATE column and TIME column, the contents of these three tables are identical. Differences in the contents of similar tables are explained in the column descriptions.

The DATE and TIME information are stored in the standard DB2 format and displayed in the local format.

Control tables

The control tables are created during installation of the Tivoli Decision Support for z/OS base. The tables control results returned by some log collector functions.

Control tables appear in the tables list in the administration dialog.

DAY_OF_WEEK

This control table defines the day type to be returned by the DAYTYPE function for each day of the week. The day type is used as a key in the PERIOD_PLAN and SCHEDULE control tables.

Column name		Data type	Description
DAY_OF_WEEK	K	SMALLINT	Day of week number, 1 through 7 (Monday through Sunday).
DAY_TYPE		CHAR(8)	Day type for the day of week.

Example of table contents

```
DAY
OF DAY
WEEK TYPE
-----
```

Control tables

1 MON
 2 TUE
 3 WED
 4 THU
 5 FRI
 6 SAT
 7 SUN

PERIOD_PLAN

This control table defines the periods to be returned by the PERIOD function, which is described in the *Language Guide and Reference*. A period plan defines the partition of a day into periods (such as shifts) for each day type defined by the DAY_OF_WEEK and SPECIAL_DAY control tables.

Column name		Data type	Description
PERIOD_PLAN_ID	K	CHAR(8)	You can have different sets of period names for different systems. Each application normally uses a system ID from the log to match this field, for example the MVS system ID for an MVS performance application. Specify % for the rows that specify your default set of period names. This can contain global search characters.
DAY_TYPE	K	CHAR(8)	Day type the period applies to. This can be any of the day types specified in the DAY_OF_WEEK and SPECIAL_DAY control tables.
START_TIME	K	TIME	Time when the period starts.
END_TIME		TIME	Time when the period ends.
PERIOD_NAME		CHAR(8)	Name of the period.

Example of table contents

```

PERIOD
PLAN   DAY   START   END   PERIOD
ID     TYPE  TIME    TIME  NAME
-----
%      MON    00.00.00 08.00.00 NIGHT
%      MON    08.00.00 17.00.00 PRIME
%      MON    17.00.00 24.00.00 NIGHT
%      TUE    00.00.00 08.00.00 NIGHT
%      TUE    08.00.00 17.00.00 PRIME
%      TUE    17.00.00 24.00.00 NIGHT
%      WED    00.00.00 08.00.00 NIGHT
%      WED    08.00.00 17.00.00 PRIME
%      WED    17.00.00 24.00.00 NIGHT
%      THU    00.00.00 08.00.00 NIGHT
%      THU    08.00.00 17.00.00 PRIME
%      THU    17.00.00 24.00.00 NIGHT
%      FRI    00.00.00 08.00.00 NIGHT
%      FRI    08.00.00 17.00.00 PRIME
%      FRI    17.00.00 24.00.00 NIGHT
%      SAT    00.00.00 24.00.00 WEEKEND
%      SUN    00.00.00 24.00.00 WEEKEND
%      HOLIDAY 00.00.00 24.00.00 HOLIDAY
  
```

SCHEDULE

This control table defines the schedules to be returned by the APPLY SCHEDULE function. A schedule is a time period when a resource is planned to be up; it is used in availability calculations.

Column name		Data type	Description
SCHEDULE_NAME	K	CHAR(8)	Name of the schedule. By giving different names to schedules, you can have different schedules for the various systems or resources. The AVAILABILITY_PARM table controls which schedule name to use for a resource.
DAY_TYPE	K	CHAR(8)	Day type the schedule applies to. This can be any of the day types specified in the DAY_OF_WEEK and SPECIAL_DAY control tables.
START_TIME	K	TIME	Time when the schedule starts.
END_TIME		TIME	Time when the schedule ends.

Example of table contents

SCHEDULE NAME	DAY TYPE	START TIME	END TIME
STANDARD	MON	08.00.00	17.00.00
STANDARD	TUE	08.00.00	17.00.00
STANDARD	WED	08.00.00	17.00.00
STANDARD	THU	08.00.00	17.00.00
STANDARD	FRI	08.00.00	17.00.00
STANDARD	SAT	00.00.00	00.00.00
STANDARD	SUN	00.00.00	00.00.00
STANDARD	HOLIDAY	00.00.00	00.00.00

SPECIAL_DAY

This control table defines the day type to be returned by the DAYTYPE function for special dates such as holidays. The day type is used as a key in the PERIOD_PLAN and SCHEDULE control tables.

Column name		Data type	Description
DATE	K	DATE	Date to be defined as special day.
DAY_TYPE		CHAR(8)	Day type for the date; for example, HOLIDAY.

Example of table contents

DATE	DAY TYPE
1999-12-25	HOLIDAY
2000-01-01	HOLIDAY

AGGR_VALUE

This table is to be used to assign a default value to a key field if it is not required in the aggregation. If a record is found in the AGGR_VALUE for a particular table and column, then the default value is used in the aggregation. This has the potential to reduce the number of rows collected for that particular table.

Column name		Data type	Description
AGGR_TABLE	K	CHAR(18)	Name of TDS table.
AGGR_COLUMN	K	CHAR(18)	Name of TDS column.
AGGR_DEF_VALUE		CHAR(16)	Default value to assign to field.

Control tables

Example of table contents

AGGR TABLE	AGGR COLUMN	AGGR DEF VALUE
DB2_PACKAGE_H	CORRELATION_ID	\$USER
DB2_PACKAGE_H	PRIMARY_AUTH_ID	\$USER

CICS control tables

The CICS control tables are created during installation of the Tivoli Decision Support for z/OS base. The tables control results returned by some log collector functions during CICS log data collection.

CICS control tables appear in the tables list in the administration dialog.

CICS_DICTIONARY

This control table is used during CICS log data collection. The CICS record procedure, DRL2CICS, uses CICS_DICTIONARY to store the latest dictionary record processed for each unique combination of MVS_SYSTEM_ID, CICS_SYSTEM_ID, CLASS and VERSION. For more information, refer to the *CICS Performance Feature Guide and Reference*, SH19-6820.

Column name		Data type	Description
MVS_SYSTEM_ID	K	CHAR(4)	MVS system ID. From SMFMNSID (V3) or SMFSID (V2).
CICS_SYSTEM_ID	K	CHAR(8)	CICS generic ID. This is the VTAM [®] application identifier for the CICS system that produced the dictionary. From SMFMNPRN (V3) or SMFPSPRN (V2).
CLASS	K	SMALLINT	Monitoring class. This is 2 for accounting (CICS/MVS V2 only), 3 for performance data, and 4 for exception data (CICS/MVS V2 only). From SMFMNCL (V3) or MNSEGCL (V2).
VERSION	K	SMALLINT	Version of the CICS system that produced the dictionary. This is 2 for CICS/MVS (V2) and 3 for CICS/ESA (V3). Set by DRL2CICS based on SMFMNSTY (V3) or SMFSTY (V2).
FIELD_NO	K	SMALLINT	Assigned connector for this dictionary entry (CMODCONN). This is also the index to the dictionary entry array.
CICS_VER	K	CHAR(4)	CICS version and release that created this dictionary (from the field SMFMNVRN). EX. 0410.
DICTIONARY_ENTRY_ID		CHAR(12)	Dictionary entry ID. It is made up of the CMODNAME, CMODTYPE and CMODIDNT fields in the dictionary entry. It is used to uniquely identify each dictionary entry.
OUTPUT_LENGTH		SMALLINT	Field length for matching DICTIONARY_ENTRY_ID in CICS_FIELD. It is used for building the output record.
OUTPUT_OFFSET		SMALLINT	Field offset for matching DICTIONARY_ENTRY_ID in CICS_FIELD. It is used for building the output record.
USED		CHAR(8)	A flag indicating (if = Y) that this dictionary entry has been updated with field length and offset data from a matching DICTIONARY_ENTRY_ID in CICS_FIELD.

CICS_FIELD

This control table is used during CICS log data collection. The CICS record procedure, DRL2CICS, uses CICS_FIELD to store field lengths and offsets for

dictionary fields described in “CICS_DICTIONARY” on page 246. For more information, refer to the *CICS Performance Feature Guide and Reference* CICS Performance Feature Guide and Reference.

Column name		Data type	Description
CLASS	K	SMALLINT	CMF record class. 2 for accounting (CICS/MVS V2 only), 3 for performance data (transaction and global (CICS/MVS V2 only)) and 4 for exception data (CICS/MVS V2 only).
DICTIONARY_ENTRY_ID	K	CHAR(12)	This is the dictionary entry ID. It is made up of the CMODNAME, CMODTYPE and CMODIDNT fields in the dictionary entry. It is used to uniquely identify each dictionary entry.
FIRST_CICS_VER	K	CHAR(4)	This is first version of CICS that introduced this CMODTYPE and CMODIDNT with these attributes. This allows multiple versions of the same key as many fields were changed with CICS TS 3.2
OUTPUT_LENGTH		SMALLINT	This is the field length that is used to build the output record.
OUTPUT_OFFSET		INTEGER	This is the field offset that is used to build the output record. This offset should match the SMF_CICS_T, _G, _A, _E2 record definitions.

Common data tables

These tables are ordinary data tables that are used by many components. They are provided with the Tivoli Decision Support for z/OS base, but not created until the installation of the first component that uses them.

Naming standard for common data tables

Names of Tivoli Decision Support for z/OS common data tables are in this format:

content_suffix

where:

- *content* is a description (for example, AVAILABILITY for system and resource availability data).
- *suffix* indicates the summarization level of the data in the table (for example, AVAILABILITY_D for availability data summarized by day).

A common table name can have these summarization-level suffixes:

- _T The table holds nonsummarized data (**timestamped** data).
- _D The table holds data summarized by **day**.
- _W The table holds data summarized by **week**.
- _M The table holds data summarized by **month**.

AVAILABILITY_D, _W, _M

These tables provide daily, weekly, and monthly statistics on the availability of systems and subsystems. They contain consolidated data from the AVAILABILITY_T table.

The default retention periods for these tables are:

AVAILABILITY_D

90 days

AVAILABILITY_W

400 days

AVAILABILITY_M

800 days

Common data tables

Column name		Data type	Description
DATE	K	DATE	Date that the availability data applies to. For the _W table, this is the date of the first day of the week. For the _M table, this is the date of the first day of the month.
SYSTEM_ID	K	CHAR(8)	System ID such as an MVS or VM system ID.
AREA	K	CHAR(8)	Major area the resource is related to, such as MVS or NETWORK.
RESOURCE_TYPE	K	CHAR(8)	Resource type.
RESOURCE_NAME	K	CHAR(8)	Resource name.
RESOURCE_GROUP	K	CHAR(8)	Resource group.
AVAIL_OBJ_PCT		DECIMAL(4,1)	Availability objective for the resource, in percent. This is from the column AVAIL_OBJ_PCT in the AVAILABILITY_PARM lookup table. This value should be compared with the actual availability, which is calculated as: 100*UP_IN_SCHEDULE/SCHEDULE_HOURS.
MEASURED_HOURS		FLOAT	Number of hours measured.
SCHEDULE_DAYS		SMALLINT	Number of days during the week or month that the resource was scheduled to be up. This column is only present in the _W and _M tables.
SCHEDULE_HOURS		FLOAT	Number of hours the resource was scheduled to be up.
STARTS		SMALLINT	Number of times the resource was started.
STARTS_IN_SCHEDULE		SMALLINT	Number of times the resource was started within the schedule.
STOPS		SMALLINT	Number of times the resource was stopped.
STOPS_IN_SCHEDULE		SMALLINT	Number of times the resource was stopped within the schedule.
UP_HOURS		FLOAT	Number of hours the resource was up.
UP_IN_SCHEDULE		FLOAT	Number of hours the resource was up within the schedule.

AVAILABILITY_T

This table provides detailed availability data about the system as a whole and all its subsystems. The data comes from many different sources. For every resource tracked, this table contains one row for each time interval with a different status.

The default retention period for this table is 10 days.

Column name		Data type	Description
SYSTEM_ID	K	CHAR(8)	System ID such as an MVS or VM system ID.
AREA	K	CHAR(8)	Major area the resource is related to, such as MVS or NETWORK.
RESOURCE_TYPE	K	CHAR(8)	Resource type.
RESOURCE_NAME	K	CHAR(8)	Resource name.
RESOURCE_GROUP	K	CHAR(8)	Resource group.

Column name		Data type	Description
INTERVAL_TYPE	K	CHAR(3)	Interval type. Possible values are: ==, =, = , = , XXX, XX, XX , X , and blank, where: = Indicates that the resource is up (available) X Indicates that the resource is down Indicates an interval start or end blank Means that the status is unknown
START_TIME	K	TIMESTAMP	Start time of the interval.
END_TIME		TIMESTAMP	End time of the interval.
QUIET_INTERVAL_SEC		INTEGER	Number of seconds after the interval end that the resource is expected to remain in the same status. If another interval with a start time within this range appears, the two intervals are merged.

EXCEPTION_T

This table provides a list of exceptions that have occurred in the system and require attention. The data comes from many different sources.

The layout of this table cannot be changed by the user.

The default retention period for this table is 14 days.

Column name		Data type	Description
DATE	K	DATE	Date when the exception occurred.
TIME	K	TIME	Time when the exception occurred.
SYSTEM_ID	K	CHAR(8)	System where the exception occurred.
AREA	K	CHAR(8)	Major area the exception is related to, such as MVS or NETWORK.
EXCEPTION_ID	K	VARCHAR(18)	Short description of the exception type. This can be used to count the number of exceptions of different types.
RESOURCE_NAME1	K	CHAR(8)	Name of the first resource that the exception is related to.
RESOURCE_NAME2	K	CHAR(8)	Name of the second resource that the exception is related to.
DATE_GENERATED		DATE	Date when the problem was recorded in the Information/Management database. This is null if no problem record has been generated.
EXCEPTION_DESC		VARCHAR(45)	Text that describes the exception, in any format.
PROBLEM_FLAG		CHAR(1)	Controls whether a problem record should be automatically generated for the exception. This can be Y (generate a problem record) or N (do not generate a problem record).
PROBLEM_NUMBER		CHAR(8)	The Information/Management problem-record number. This is null if no problem record has been generated.
SEVERITY		CHAR(2)	Severity of the problem. This is user-defined.
TRANSACTION_NUMBER		INTEGER	Transaction identifier number.
TRANSACTION_CHAR		CHAR(4)	Transaction number in character format. (in some special cases CICS system tasks are identified as III, JBS, J01-J99, TCB.)
PROGRAM_NAME		CHAR(8)	Name of the program.

Common data tables

MIGRATION_LOG

This table holds information on what migration jobs have been run, and the results of each step.

The layout of this table cannot be changed by the user.

The default retention period for this table is 14 days.

Column name		Data type	Description
JOB_NAME	K	CHAR(8)	Migration job name.
STEP_NO	K	INTEGER	Step number of job.
START_DATE	K	DATE	Start date of job.
START_TIME	K	TIME	Start time of job.
STEP_NAME		CHAR(30)	Step name of job.
RETURN_CODE		INTEGER	Step status code.
COMPLETED_CODE		CHAR	Y – Completed successfully U – Abend
END_DATE		DATE	End date of last migration step.
END_TIME		TIME	End time of last migration step.

Common lookup tables

These tables are ordinary lookup tables that are used by many components. They are provided with the Tivoli Decision Support for z/OS base, but not created until the installation of the first component that uses them.

AVAILABILITY_PARM

This lookup table sets availability parameters. It contains the schedule names and availability objectives to use for the different resources in the system. Its values are used in the AVAILABILITY_D, _W, and _M tables.

Column name		Data type	Description
SYSTEM_ID	K	CHAR(8)	System ID associated with the resource. This can contain global search characters.
AREA	K	CHAR(8)	Major area that the resource is related to, such as MVS or NETWORK. This can contain global search characters.
RESOURCE_TYPE	K	CHAR(8)	Resource type. This can contain global search characters.
RESOURCE_NAME	K	CHAR(8)	Resource name. This can contain global search characters.
RESOURCE_GROUP	K	CHAR(8)	Resource group. This can contain global search characters.
AVAIL_OBJ_PCT		DECIMAL(4,1)	Availability objective for the resource, in percent.
SCHEDULE_NAME		CHAR(8)	Schedule name to use for the resource.

Example of table contents

SYSTEM ID	AREA	RESOURCE TYPE	RESOURCE NAME	RESOURCE GROUP	SCHEDULE NAME	AVAIL OBJ PCT
-----	-----	-----	-----	-----	-----	-----

... % % % % % STANDARD 95.0

USER_GROUP

This lookup table groups the users of the system into user groups. The values are used in many tables. You can also assign division and department names to the user groups; however, the names are left blank in the predefined tables.

Column name		Data type	Description
SYSTEM_ID	K	CHAR(8)	System ID such as an MVS or VM system ID. This can contain global search characters.
SUBSYSTEM_ID	K	CHAR(8)	Subsystem ID such as TSO or a CICS* system ID. This can contain global search characters. This is not used in the predefined tables.
USER_ID	K	CHAR(8)	User ID of the user to be grouped. This can contain global search characters.
DEPARTMENT		CHAR(8)	Department that the user belongs to. This is not used in the predefined tables.
DIVISION		CHAR(8)	Division that the user belongs to. This is not used in the predefined tables.
GROUP_NAME		CHAR(8)	Name of the group that the user belongs to.

Example of table contents

SYSTEM ID	SUBSYSTEM ID	USER ID	DIVISION	DEPARTMENT	GROUP NAME
-----	-----	-----	-----	-----	-----
*	*	USER1			GROUP1
*	*	USER2			GROUP2
...					

Common lookup tables

Chapter 13. Sample components

This appendix describes the Sample component, the only component shipped with the Tivoli Decision Support for z/OS base product.

The sample component consists of:

- A sample log and record definition
- Three sample tables with update definitions
- Three sample reports
- A log data set with sample data that can be collected

Figure 106 shows an overview of the flow of data from the sample log data set, DRLSAMPL (in the DRLxxx.SDRLDEFS library), through the Sample component of Tivoli Decision Support for z/OS, and finally into reports.

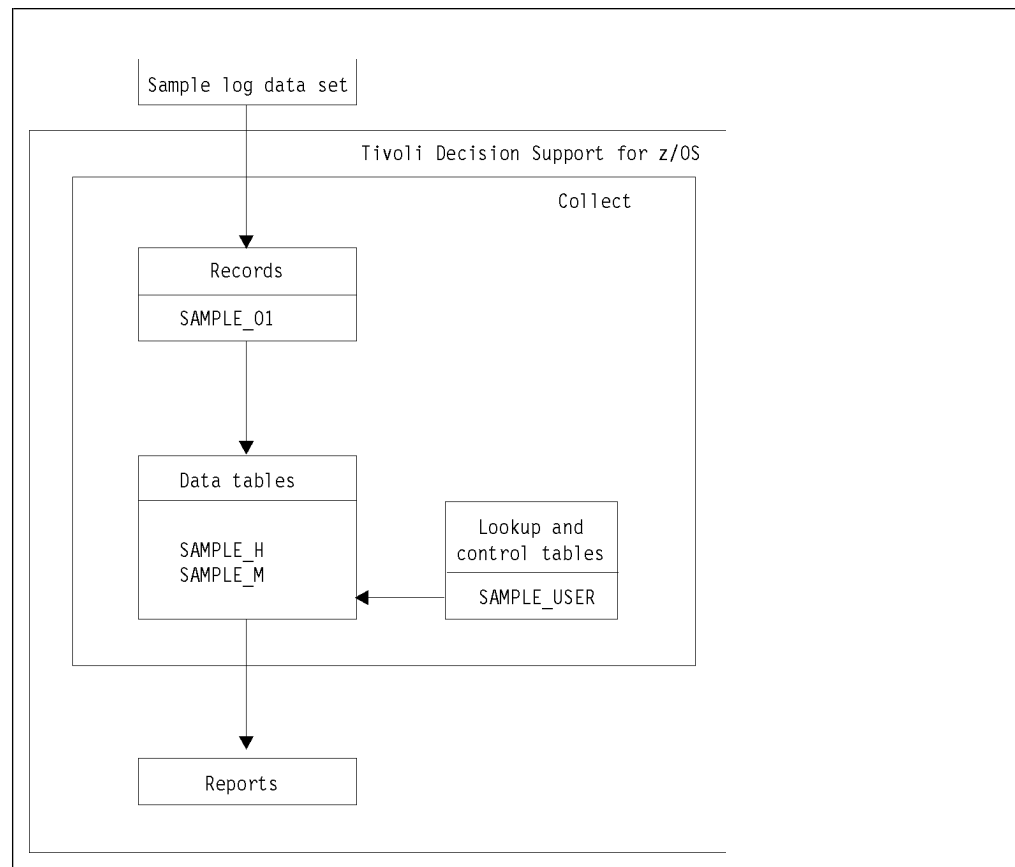


Figure 106. Sample data flow

Sample component

You can use the Sample component for testing the installation of the base product or to demonstrate Tivoli Decision Support for z/OS.

Data tables

SAMPLE_H, _M data tables

These tables provide hourly and monthly sample data.

Column name		Data type	Description
DATE	K	DATE	Date. For the _M table, this is the date of the first day of the month. From S01DATE.
TIME	K	TIME	Time rounded down to the nearest hour. This applies only to the _H table. From S01TIME.
SYSTEM_ID	K	CHAR(4)	System ID. From S01SYST.
DEPARTMENT_NAME	K	CHAR(8)	Department name. From DEPARTMENT_NAME in the SAMPLE_USER lookup table. This is derived using field S01USER from the record as key.
USER_ID	K	CHAR(8)	User ID. From S01USER.
CPU_SECONDS		FLOAT	Total processor time, in seconds. Calculated as the sum of S01CPU/100.0.
PAGES_PRINTED		INTEGER	Number of pages printed. This is the sum of S01PRNT.
RESPONSE_SECONDS		INTEGER	Total response time, in seconds. This is the sum of S01RESP.
TRANSACTIONS		INTEGER	Number of transactions. This is the sum of S01TRNS.

SAMPLE_USER lookup table

This lookup table assigns department names to users.

Column name		Data type	Description
USER_ID	K	CHAR(8)	User ID
DEPARTMENT_NAME		CHAR(8)	Department name

Example of table contents

USER ID	DEPARTMENT NAME
-----	-----
ADAMS	App1 Dev
GEYER	Finance
GOUNOT	Retail
HAAS	Finance
JONES	App1 Dev
KWAN	Marketng
LEE	Manufact
LUTZ	Manufact
MARINO	Retail
MEHTA	Manufact
PARKER	Finance
PEREZ	Retail

Sample components reports

In the report descriptions that follow, this information is included:

Heading

The title of the report.

Introduction

A brief introduction to the purpose of the report.

Report ID

Tivoli Decision Support for z/OS assigns each report a unique report identifier. Each report ID consists of SAMPLE and a sequential number, such as SAMPLE01.

Report group

To make it easier to find reports, Tivoli Decision Support for z/OS organizes reports into report groups, which correspond to feature components. Sample component reports belong to the Sample report group.

Source

Each Sample report contains information adapted from either the SAMPLE_H or SAMPLE_M source tables.

Attributes

Attributes are keys that you can use to search for a particular report. The Sample component reports each have one attribute, Sample.

Variables

Each report has several variables associated with it. When you select a report to display, Tivoli Decision Support for z/OS prompts you for the variables listed in the description.

Example report

Each example illustrates a typical report.

Column descriptions

Column descriptions identify the information contained within the report, in detail. If the column contains a calculated value, the formula used for the calculation is included.

Sample Report 1

This surface chart shows the processor time consumed by different projects. It gives an hourly profile for an average day.

This information identifies the report:

Report ID

SAMPLE01

Report group

Sample Reports

Source

SAMPLE_H

Chart format

DRLGSURF

Attributes

Sample

Variables

System ID

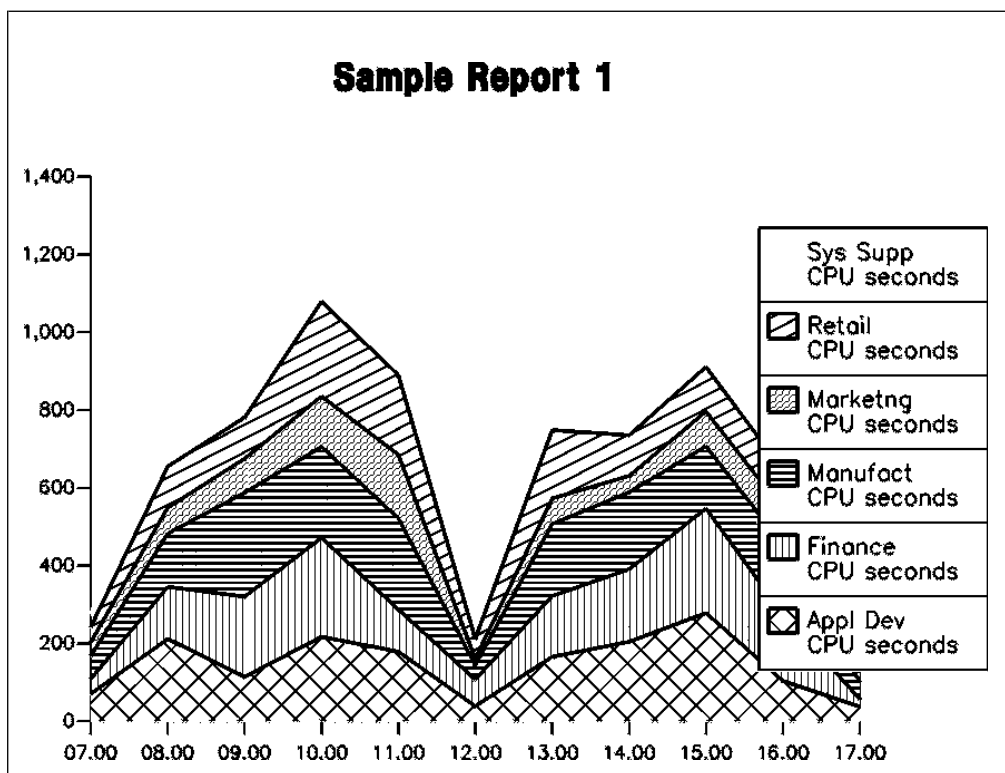


Figure 107. Sample Report 1

The report contains this information:

Horizontal axis

Hour, in the format *hh.mm*

Vertical axis

Processor time, in seconds

Legend

Department name

Sample Report 2

This report shows the resources consumed by each user and department.

This information identifies the report:

Report ID

SAMPLE02

Report group

Sample Reports

Source

SAMPLE_M

Attributes

Sample

Variables

From_month, To_month, System_ID

Sample Report 2

Month start date	Department name	User ID	Trans- actions	Average response seconds	CPU seconds	Pages printed
2000-01-01	Appl Dev	ADAMS	1109	3.84	244.13	821
		JONES	1138	3.40	228.79	1055
		SMITH	870	4.27	183.03	864
		*	3117	3.84	655.95	2740
	Finance	GEYER	509	4.29	115.97	529
		HAAS	786	3.56	137.48	648
		PARKER	462	6.79	171.51	704
		SPENCER	800	3.33	172.82	640
		*	2557	4.50	597.78	2521
	:					
			36396	4.03	7868.97	38711

Tivoli Decision Support for z/OS Report: SAMPLE02

Figure 108. Sample Report 2

The columns in this report contain this information:

Month start date

Date of the first day in the month.

Department name

Name of the department that the user belongs to.

User ID

ID of the user.

Transactions

Number of transactions run by the user.

Average response seconds

The average response time, in seconds for all transactions. Calculated as RESPONSE_SECONDS/TRANSACTIONS.

CPU seconds

Number of processor seconds consumed.

Pages printed

Number of pages printed.

Sample Report 3

This bar chart shows the processor time consumed by each project during the selected time period, sorted as a toplist.

This information identifies the report:

Report ID

SAMPLE03

Report group

Sample Reports

Source

SAMPLE_M

Chart format

DRLGHORB

Attributes

Sample

Reports

Variables

From_date, To_date, System_ID

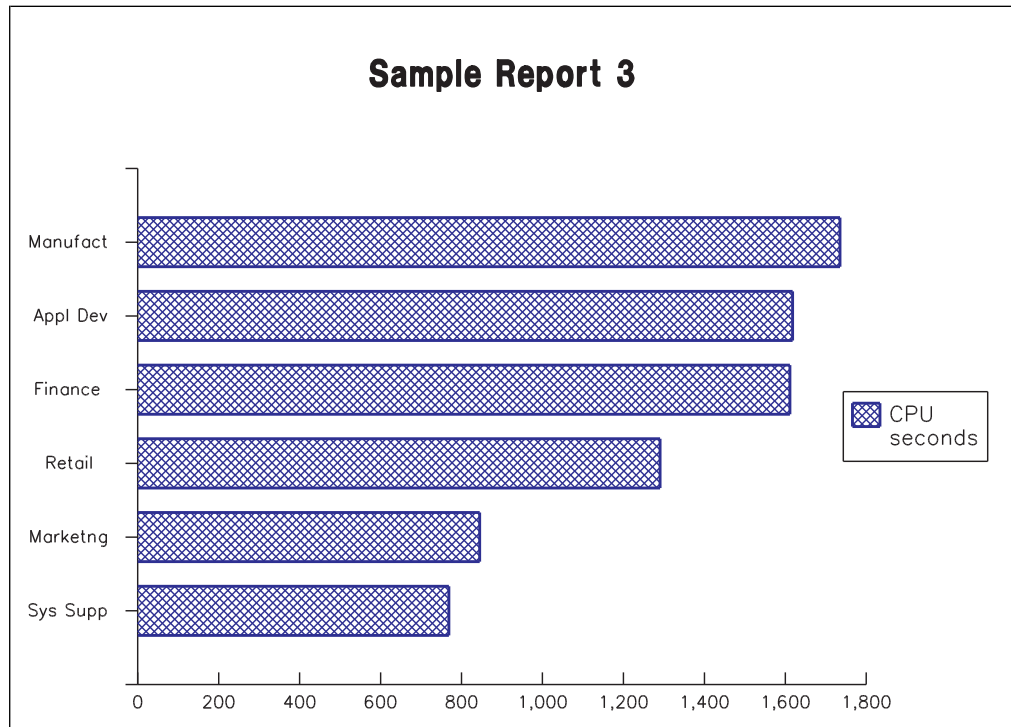


Figure 109. Sample Report 3

The report contains this information:

Horizontal axis

Processor time, in seconds

Vertical axis

Department name

Chapter 14. Record definitions supplied with Tivoli Decision Support for z/OS

In addition to the records used by the components, the Tivoli Decision Support for z/OS base product contains definitions of many records. This chapter lists all the records defined by the base product, except for those built by Tivoli Decision Support for z/OS exits and utilities.

SMF records

Record name	Member name	Description
SMF_000	DRLRS000	IPL
SMF_002	DRLRS002	Dump header
SMF_003	DRLRS003	Dump trailer
SMF_004	DRLRS004	Step termination
SMF_005	DRLRS005	Job termination
SMF_006	DRLRS006	JES2/JES3/PSF/External writer
SMF_007	DRLRS007	Data lost
SMF_008	DRLRS008	I/O configuration
SMF_009	DRLRS009	VARY device ONLINE
SMF_010	DRLRS010	Allocation recovery
SMF_011	DRLRS011	VARY device OFFLINE
SMF_014	DRLRS014	INPUT or RDBACK data set activity
SMF_015	DRLRS015	OUTPUT, UPDAT, INOUT, or OUTIN data set activity
SMF_016	DRLRS016	DFSORT statistics
SMF_017	DRLRS017	Scratch data set status
SMF_018	DRLRS018	Rename data set status
SMF_019	DRLRS019	Direct access volume
SMF_020	DRLRS020	Job initiation
SMF_021	DRLRS021	Error statistics by volume
SMF_022	DRLRS022	Configuration
SMF_023	DRLRS023	SMF status
SMF_024	DRLRS024	JES2 spool offload
SMF_025	DRLRS025	JES3 device allocation
SMF_026	DRLRS026	JES2/JES3 job purge
SMF_028	DRLRS028	NPM statistics. SMF_028 maps all subtypes of SMF type 28. To improve performance, the subtypes used by Tivoli Decision Support for z/OS are mapped with special record definitions (SMF_028_xxx). Note that SMF_028 cannot be used together with these definitions because each log record can be mapped by only one record definition.
SMF_028_NTRI	DRLRSNTR	NPM NTRI statistics
SMF_028_TRANSIT	DRLRSNTT	NPM transit time statistics

SMF records

Record name	Member name	Description
SMF_028_TRANS_SUM	DRLRSNT1	NPM Transit Time summary statistics
SMF_028_X25	DRLRSX25	NPM X25 statistics
SMF_028_PU	DRLRSNPU	NPM PU statistics
SMF_028_NPM	DRLRSNPM	NPM internal statistics
SMF_028_LINE	DRLRSNLI	NPM line statistics
SMF_028_NEO	DRLRSNEO	NPM NEO statistics
SMF_028_NCP	DRLRSNCP	NPM NCP statistics
SMF_028_LAN	DRLRSLAN	NPM LAN statistics
SMF_028_VTAM	DRLRSVTM	NPM VTAM statistics
SMF_030	DRLRS030	Common address space work
SMF_031	DRLRS031	TIOC initialization
SMF_032	DRLRS032	TSO user work accounting
SMF_033	DRLRS033	APPC/MVS TP accounting
SMF_034	DRLRS034	TS-step termination
SMF_035	DRLRS035	LOGOFF
SMF_036	DRLRS036	ICF catalog
SMF_037_HW	DRLRS037	NetView Hardware Monitor
SMF_037_VPD	DRLRSVPD	Network configuration (VPD)
SMF_039_1_TO_7	DRLRS039	NetView Session Monitor, SMF 39, subtypes 1 to 7
SMF_039_8	DRLRS039	NetView Session Monitor, SMF 39, subtype 8
SMF_040	DRLRS040	Dynamic DD
SMF_041	DRLRS041	Data-in-virtual Access/Unaccess
SMF_042_1	DRLRS042	BMF performance statistics
SMF_042_2	DRLRS042	DFP cache control unit statistics
SMF_042_3	DRLRS042	DFP SMS configuration statistics
SMF_042_5	DRLRSX42	DFSMS storage class statistics
SMF_042_6	DRLRSX42	DFSMS Data Set statistics
SMF_042_14	DRLRADSM	ADSTAR Distributed Storage Manager (ADSM) server statistics
SMF_042_11	DRLRSX42	DFP Extended Remote Copy (XRC) session statistics
SMF_043_2	DRLRS043	JES2 start
SMF_043_5	DRLRS043	JES3 start
SMF_045_2	DRLRS045	JES2 withdrawal
SMF_045_5	DRLRS045	JES3 stop
SMF_047_2	DRLRS047	JES2 SIGNON/start line (BSC only)
SMF_047_5	DRLRS047	JES3 SIGNON/start line/LOGON
SMF_048_2	DRLRS048	JES2 SIGNOFF/stop line (BSC only)
SMF_048_5	DRLRS048	JES3 SIGNOFF/stop line/LOGOFF
SMF_049_2	DRLRS049	JES2 integrity (BSC only)
SMF_049_5	DRLRS049	JES3 integrity
SMF_050	DRLRS050	ACF/VTAM* tuning statistics
SMF_052	DRLRS052	JES2 LOGON/start line (SNA only)

Record name	Member name	Description
SMF_053	DRLRS053	JES2 LOGOFF/start line (SNA only)
SMF_054	DRLRS054	JES2 integrity (SNA only)
SMF_055	DRLRS055	JES2 network SIGNON
SMF_056	DRLRS056	JES2 network integrity
SMF_057_2	DRLRS057	JES2 network SYSOUT transmission
SMF_057_5	DRLRS057	JES3 networking transmission
SMF_058	DRLRS058	JES2 network SIGNOFF
SMF_059	DRLRS059	MVS/BDT file-to-file transmission
SMF_060	DRLRS060	VSAM volume data set updated
SMF_061	DRLRS061	ICF define activity
SMF_062	DRLRS062	VSAM component or cluster opened
SMF_063	DRLRS063	VSAM entry defined
SMF_064	DRLRS064	VSAM component or cluster status
SMF_065	DRLRS065	ICF delete activity
SMF_066	DRLRS066	ICF alter activity
SMF_067	DRLRS067	VSAM entry delete
SMF_068	DRLRS068	VSAM entry renamed
SMF_069	DRLRS069	VSAM data space defined, extended, or deleted
SMF_070	DRLRS070	RMF™ CPU activity
SMF_071	DRLRS071	RMF paging activity
SMF_072_1	DRLRS072	RMF workload activity
SMF_072_2	DRLRSX72	RMF storage data
SMF_072_3	DRLRS072	RMF goal mode workload activity
SMF_072_4	DRLRSX72	RMF goal mode delay and storage frame data
SMF_073	DRLRS073	RMF channel path activity
SMF_074_1	DRLRS074	RMF device activity
SMF_074_2	DRLRS074	RMF XCF activity
SMF_074_3	DRLRSX74	RMF Device OMVS activity
SMF_074_4	DRLRSX74	RMF XES/CF activity
SMF_074_6	DRLRX74	File system statistics
SMF_075	DRLRS075	RMF page/swap data set activity
SMF_076	DRLRS076	RMF trace activity
SMF_077	DRLRS077	RMF enqueue activity
SMF_078_1	DRLRS078	RMF I/O queueing activity for the 308x, 908x, and 4381 processors
SMF_078_2	DRLRS078	RMF virtual storage activity
SMF_078_3	DRLRS078	RMF I/O queueing activity for the 3090, 9021, 9121, and 9221 processors
SMF_079	DRLRS079	RMF Monitor II activity
SMF_080	DRLRS080	RACF processing
SMF_081	DRLRS081	RACF initialization
SMF_082_1	DRLRS082	PCF record

SMF records

Record name	Member name	Description
SMF_082_2	DRLRS082	CUSP record
SMF_083	DRLRS083	RACF audit record for data sets
SMF_084_1	DRLRS084	JMF - FCT analysis
SMF_084_2	DRLRS084	JMF - FCT summary and highlights
SMF_084_3	DRLRS084	JMF - spool data management
SMF_084_4	DRLRS084	JMF - resqueue cellpool, JCT and control block utilization
SMF_084_5	DRLRS084	JMF - job analysis
SMF_084_6	DRLRS084	JMF - JES3 hot spot analysis
SMF_084_7	DRLRS084	JMF - JES internal reader DSP analysis
SMF_084_8	DRLRS084	JMF - JES3 SSI response time analysis
SMF_084_9	DRLRS084	JMF - JES3 SSI destination queue analysis
SMF_085	DRLRS085	OAM record
SMF_088	DRLRS088	System logger
SMF_089	DRLRS089	Product Usage Data
SMF_090	DRLRS090	System status
SMF_092	DRLRS092	z/OS UNIX activity
SMF_094	DRLRS094	3494, 3495 Tape library data server statistics
SMF_099	DRLRS099	SMS System Resource Manager decisions
SMF_100_0	DRLRS100	DB2 statistics, system services
SMF_100_1	DRLRS100	DB2 statistics, database services
SMF_100_2	DRLRS100	DB2 statistics, dynamic ZPARMs
SMF_100_3	DRLRS100	DB2 statistics, Buffer, Manager Group Buffer Pool
SMF_101	DRLRS101	DB2 accounting
SMF_101_1	DRLRS101	DB2 accounting, Packages extension
SMF_102	DRLRS102	DB2 system initialization parameters
SMF_110_0	DRLRS110	CICS/ESA journaling record
SMF_110_0_V2	DRLRS110	CICS/MVS monitoring record
SMF_110_1	DRLRS110	CICS/ESA monitoring record
SMF_110_1_1	DRLRS110	CICS/TS <3.2 record
SMF_110_1_5	DRLR110T	CICS transaction resource - expanded
SMF_110_2	DRLR1102	CICS/ESA and CICS/TS statistics record
SMF_110_3	DRLRS1103	CICS/TS statistics record
SMF_110_4	DRLR1103	CICS/TS CF statistics record
SMF_110_5	DRLR1103	CICS/TS NC statistics record
SMF_110_1_C	DRLRS110	CICS/TS 3.2+ - may be compressed
SMF_110_1_CO	DRLRS110	CICS/TS 3.2+ - expanded
SMF_110_E	DRLRS110	CICS/ESA exception record - expanded
SMF_112_203_C	DRLRS112	OMEGAMON® XE for CICS file and database usage – compressed
SMF_112_203	DRLRS112	OMEGAMON XE for CICS file and database usage – expanded
SMF_114_1	DRLRS114	System Automation Tracking
SMF_115	DRLRS115	WebSphere MQ for z/OS statistics

Record name	Member name	Description
SMF_116	DRLRS116	WebSphere MQ for z/OS statistics
SMF_117	DRLRS117	Websphere Message Broker
SMF_118_1	DRLRS118	TCP/IP API calls record
SMF_118_3	DRLRS118	TCP/IP FTP client calls record
SMF_118_4	DRLRS118	TCP/IP TELNET client calls record
SMF_118_20	DRLRS118	TCP/IP TELNET server record
SMF_118_5	DRLRS118	TCP/IP general statistics record
SMF_118_70	DRLRS118	TCP/IP FTP server record
SMF_119_1	DRLRS119	TCP connection initiation
SMF_119_2	DRLRS119	TCP connection termination
SMF_119_3	DRLRS119	FTP client transfer completion
SMF_119_4	DRLRS119	TCP/IP Profile Information record
SMF_119_5	DRLRS119	TCP/IP statistics
SMF_119_6	DRLRS119	Interface statistics
SMF_119_7	DRLRS119	Server port statistics
SMF_119_8	DRLRS119	TCP/IP stack start/stop
SMF_119_10	DRLRS119	UDP socket close
SMF_119_20	DRLRS119	TN3270 server SNA session initiation
SMF_119_21	DRLRS119	TN3270 server SNA session termination
SMF_119_22	DRLRS119	TSO telnet client connection initiation
SMF_119_23	DRLRS119	TSO telnet client connection termination
SMF_119_70	DRLRS119	FTP server transfer completion
SMF_119_72	DRLRS119	FTP server logon failure
SMF_119_73	DRLRS119	IPSec IKE Tunnel Activation/Refresh record
SMF_119_74	DRLRS119	IPSec IKE Tunnel Deactivation/Expire record
SMF_119_75_80	DRLRS119	IPSec Dynamic Tunnel Activation/Refresh
SMF_119_75_80	DRLRS119	IPSec Dynamic Tunnel Deactivation record
SMF_119_75_80	DRLRS119	IPSec Dynamic Tunnel Added record
SMF_119_75_80	DRLRS119	IPSec Dynamic Tunnel Removed record
SMF_119_75_80	DRLRS119	IPSec Manual Tunnel Activation record
SMF_119_75_80	DRLRS119	IPSec Manual Tunnel Deactivation record
SMF_120_1	DRLRS121	Server activity record
SMF_120_2	DRLRS122	WebSphere Application Server container activity record
SMF_120_3	DRLRS123	Server interval record
SMF_120_4	DRLRS124	WebSphere Application Server container interval record
SMF_120_5	DRLRSJWA	J2EE container activity record
SMF_120_6	DRLRSJWI	J2EE container interval record
SMF_120_7	DRLRSJWA	Web container activity record
SMF_120_8	DRLRSJWI	Web container interval record
SMF_120_9	DRLRS129	Request Activity record
SMF_120_10	DRLRS12A	Outbound Request record

SMF records

Record name	Member name	Description
SMF_120_11	DRLRSRWL	Liberty request record
SMF_120_12	DRLRSJWL	Liberty Java batch job record
SMF_123	DRLRS123	SMF HPQS statistics
SMF_194	DRLRS194	TS7700 Virtualization Engine statistics record
SMF_IXFP_01	DRLRIXFP	IXFP subsystem performance
SMF_IXFP_02	DRLRIXFP	IXFP channel interface statistics
SMF_IXFP_03	DRLRIXFP	IXFP functional device performance
SMF_IXFP_04	DRLRIXFP	IXFP device module performance
SMF_IXFP_05	DRLRIXFP	IXFP deleted data space release
SMF_IXFP_06	DRLRIXFP	IXFP snapshot event data
SMF_IXFP_07	DRLRIXFP	IXFP space utilization record
SMF_IXFP_08	DRLRIXFP	IXFP snapshot extended event data record

These records are user-defined; that is, they are not part of the standard IBM records in the range 0–127. However, they are written by IBM licensed programs.

The default record numbers are provided within parentheses.

Record name	Member name	Description
SMF_CACHE_03	DRLRS245	Cache RMF Reporter, 3990 model 03 (245)
SMF_CACHE_06	DRLRS245	Cache RMF Reporter, 3990 model 06 (245)
SMF_CACHE_13	DRLRS245	Cache RMF Reporter, 3880 model 13 (245)
SMF_CACHE_23	DRLRS245	Cache RMF Reporter, 3880 model 23 (245)
SMF_FTP	DRLRSFTP	NetView File Transfer Program (FTP) log record (252)

DFSMS/RMM records

Record name	Member name	Description
DFRMM_VOLUME	DRLRRMMV	Extract file volume record
DFRMM_RACK	DRLRRMMR	Extract file rack number record
DFRMM_SLBIN	DRLRRMMS	Extract file storage location bin record
DFRMM_PRODUCT	DRLRRMMP	Extract file product record
DFRMM_VRS	DRLRRMMK	Extract file VRS record
DFRMM_OWNER	DRLRRMMO	Extract file owner record
DFRMM_DATASET	DRLRRMMD	Extract file dataset record

IMS SLDS records

These records come from the IMS recovery log.

No reliable release indicators exist in the IMS records, so one log definition exists for each IMS release supported. The log and record names contain *Vnn* where *nn* is the IMS version and release; 71 for IMS 7.1, 81 for IMS version 8.1, 91 for IMS version 9.1, A1 for IMS version 10.1, B1 for IMS 11.1, C1 for IMS for IMS 12.1, and D1 for IMS for IMS 13.1.

The records are described in IMS mapping macros.

Record name	Member name	Description
IMS_Vnn0_01	DRLRInnS	Message Queue record (message received from a CNT)
IMS_Vnn0_02	DRLRInnS	IMS command record
IMS_Vnn0_03	DRLRInnS	Message Queue record (message received from an SMB or IMS)
IMS_Vnn0_06	DRLRInnS	IMS event accounting record
IMS_Vnn0_07	DRLRInnS	Program termination accounting record
IMS_Vnn0_08	DRLRInnS	Program schedule record
IMS_Vnn0_10	DRLRInnS	Security violation record
IMS_Vnn0_11	DRLRInnS	Start of conversation record
IMS_Vnn0_12	DRLRInnS	End of conversation record
IMS_Vnn0_13	DRLRInnS	SPA insert record
IMS_Vnn0_16	DRLRInnS	Sign on/off record
IMS_Vnn0_18	DRLRInnS	Extended checkpoint record
IMS_Vnn0_20	DRLRInnS	Database open record
IMS_Vnn0_21	DRLRInnS	Database close record
IMS_Vnn0_24	DRLRInnS	Database error record
IMS_Vnn0_30	DRLRInnS	Message queue prefix changed record
IMS_Vnn0_31	DRLRInnS	Message queue GU record
IMS_Vnn0_32	DRLRInnS	Message queue reject record
IMS_Vnn0_33	DRLRInnS	Message queue DRRN free record
IMS_Vnn0_34	DRLRInnS	Message queue cancel record
IMS_Vnn0_35	DRLRInnS	Message queue enqueue record
IMS_Vnn0_36	DRLRInnS	Message queue dequeue record
IMS_Vnn0_37	DRLRInnS	Message queue syncpoint transfer record
IMS_Vnn0_38	DRLRInnS	Message queue syncpoint fail record
IMS_Vnn0_4C	DRLRInnS	Program/Database start/stop record
IMS_Vnn0_400D	DRLRInnS	Checkpoint CCB record
IMS_Vnn0_400E	DRLRInnS	Checkpoint SPA record
IMS_Vnn0_4001	DRLRInnS	Checkpoint begin
IMS_Vnn0_4002	DRLRInnS	Checkpoint message queue record
IMS_Vnn0_4003	DRLRInnS	Checkpoint CNT record
IMS_Vnn0_4004	DRLRInnS	Checkpoint SMB record
IMS_Vnn0_4005	DRLRInnS	Checkpoint CTB record
IMS_Vnn0_4006	DRLRInnS	Checkpoint DMB record
IMS_Vnn0_4007	DRLRInnS	Checkpoint PSB record
IMS_Vnn0_4008	DRLRInnS	Checkpoint CLB record

IMS SLDS records

Record name	Member name	Description
IMS_Vnn0_4014	DRLRInnS	Checkpoint SPA QB record
IMS_Vnn0_4015	DRLRInnS	Checkpoint EQE record
IMS_Vnn0_4020	DRLRInnS	Checkpoint CIB record
IMS_Vnn0_4021	DRLRInnS	Checkpoint VTCB record
IMS_Vnn0_4070	DRLRInnS	Checkpoint MSDB begin
IMS_Vnn0_4071	DRLRInnS	Checkpoint MSDB ECNT record
IMS_Vnn0_4072	DRLRInnS	Checkpoint MSDB header
IMS_Vnn0_4073	DRLRInnS	Checkpoint MSDB pagefixed
IMS_Vnn0_4074	DRLRInnS	Checkpoint MSDB pageable
IMS_Vnn0_4079	DRLRInnS	Checkpoint MSDB end
IMS_Vnn0_4080	DRLRInnS	Checkpoint Fast Path begin
IMS_Vnn0_4081	DRLRInnS	Checkpoint Fast Path ECNT record
IMS_Vnn0_4082	DRLRInnS	Checkpoint Fast Path EMHB record
IMS_Vnn0_4083	DRLRInnS	Checkpoint Fast Path RCTE record
IMS_Vnn0_4084	DRLRInnS	Checkpoint Fast Path DMCB/DMAC record
IMS_Vnn0_4085	DRLRInnS	Checkpoint Fast Path MTO buffer record
IMS_Vnn0_4086	DRLRInnS	Checkpoint Fast Path DMHR/DEDB buffer record
IMS_Vnn0_4087	DRLRInnS	Checkpoint Fast Path ADSC record
IMS_Vnn0_4088	DRLRInnS	Checkpoint Fast Path IEEQE record
IMS_Vnn0_4089	DRLRInnS	Checkpoint Fast Path end
IMS_Vnn0_4098	DRLRInnS	Checkpoint end blocks record
IMS_Vnn0_4099	DRLRInnS	Checkpoint end queues record
IMS_Vnn0_41	DRLRInnS	Checkpoint batch record
IMS_Vnn0_42	DRLRInnS	Log buffer control record
IMS_Vnn0_43	DRLRInnS	Log dataset control record
IMS_Vnn0_45FF	DRLRInnS	End of statistics
IMS_Vnn0_450A	DRLRInnS	Statistics latch record
IMS_Vnn0_450B	DRLRInnS	Statistics dispatch storage record
IMS_Vnn0_450C	DRLRInnS	Statistics DFSCBT00 storage record
IMS_Vnn0_450D	DRLRInnS	Statistics RecAny pool record
IMS_Vnn0_450E	DRLRInnS	Statistics fixed pools storage record
IMS_Vnn0_450F	DRLRInnS	Dispatcher statistics record
IMS_Vnn0_4502	DRLRInnS	Statistics queue pool record
IMS_Vnn0_4503	DRLRInnS	Statistics format buffer pool record
IMS_Vnn0_4504	DRLRInnS	Statistics database buffer pool
IMS_Vnn0_4505	DRLRInnS	Statistics main pools record
IMS_Vnn0_4506	DRLRInnS	Statistics scheduling stats record
IMS_Vnn0_4507	DRLRInnS	Statistics logger record
IMS_Vnn0_4508	DRLRInnS	Statistics VSAM subpool record
IMS_Vnn0_4509	DRLRInnS	Statistics program isolation record
IMS_Vnn0_47	DRLRInnS	Statistics active region record

Record name	Member name	Description
IMS_Vnn0_48	DRLRInnS	OLDS padding record
IMS_Vnn0_5050	DRLRInnS	Full function database update undo/redo successful record
IMS_Vnn0_5051	DRLRInnS	Full function database update unsuccessful record
IMS_Vnn0_5052	DRLRInnS	Full function database update undo KSDS insert record
IMS_Vnn0_5501FE00	DRLRInnS	External sub-system DB2 snap in doubt record
IMS_Vnn0_56	DRLRInnS	External sub-system record
IMS_Vnn0_5901	DRLRInnS	EMH input record
IMS_Vnn0_5903	DRLRInnS	EMH output record
IMS_Vnn0_5920	DRLRInnS	Fast path MSDB change record
IMS_Vnn0_5921	DRLRInnS	Fast path DEDB area dataset open record
IMS_Vnn0_5922	DRLRInnS	Fast path DEDB area dataset close record
IMS_Vnn0_5923	DRLRInnS	Fast path DEDB area dataset status record
IMS_Vnn0_5924	DRLRInnS	Fast path DEDB area dataset EQE creation record
IMS_Vnn0_5936	DRLRInnS	EMH dequeue record
IMS_Vnn0_5937	DRLRInnS	EMH FP syncpoint record
IMS_Vnn0_5938	DRLRInnS	EMH FP syncpoint failure record
IMS_Vnn0_5950	DRLRInnS	Fast Path database update record
IMS_Vnn0_5953	DRLRInnS	Fast Path database update (utilities) record
IMS_Vnn0_5954	DRLRInnS	Fast Path database DEDB open record
IMS_Vnn0_5955	DRLRInnS	Fast Path sequential dependent syncpoint record
IMS_Vnn0_5957	DRLRInnS	Fast Path database DMAC record
IMS_Vnn0_5970	DRLRInnS	Fast Path hot standby MSDB relocation record
IMS_Vnn0_67	DRLRInnS	Communications trace, DMHR on I/O error and snap trace records
IMS_Vnn0_67FA	DRLRInnS	Trace table log record
IMS_Vnn0_7201	DRLRInnS	ETO user create record
IMS_Vnn0_7202	DRLRInnS	ETO user delete record
IMS_Vnn0_7203	DRLRInnS	ETO user modify record
IMS_Vnn0_7204	DRLRInnS	ETO lterm addition record

DCOLLECT records

These records are produced by the DFP DCOLLECT utility.

For a description of these records, refer to *z/OS DFSMS: Access Method Services for Catalog*.

Record name	Member name	Description
DCOLLECT_A	DRLRDCOA	VSAM base cluster association name
DCOLLECT_AG	DRLRDCAG	Aggregate Group information
DCOLLECT_B	DRLRDCOB	Data set backup version information
DCOLLECT_BC	DRLRDCBC	Base Configuration information
DCOLLECT_C	DRLRDCOC	DASD capacity planning information

DCOLLECT records

Record name	Member name	Description
DCOLLECT_D	DRLRDCOD	Active data set information
DCOLLECT_DC	DRLRDCDC.	Data Class construct information
DCOLLECT_DR	DRLRDCDR.	Optical Drive information
DCOLLECT_LB	DRLRDCLB.	Optical Library information
DCOLLECT_M	DRLRDCOM	Migration data set information
DCOLLECT_MC	DRLRDCMC	Management Class construct information
DCOLLECT_SC	DRLRDCSC	Storage Class construct information
DCOLLECT_SG	DRLRDCSG	Storage Group construct information
DCOLLECT_T	DRLRDCOT	Tape capacity planning information
DCOLLECT_V	DRLRDCOV	Volume information
DCOLLECT_VL	DRLRDCVL.	SMS Volume information

EREP records

For a description of these records, refer to the *Environmental Record Editing and Printing Program (EREP) User's Guide and Reference*.

Record name	Member name	Description
EREP_30	DRLRE030	DASD long outboard record
EREP_36	DRLER036	VTAM long outboard record
EREP_50	DRLER050	IPL system initialization record

Linux on zSeries records

These records are produced by the zLinux programs on your zLinux nodes.

Record name	Member name	Description
ZLINUX_CPU	DRLRZPCP	zLinux CPU performance record
ZLINUX_DISK_FS	DRLRZPDI	zLinux disk space performance record
ZLINUX_DISKIO	DRLRZPIO	zLinux disk I/O performance record
ZLINUX_PAGING	DRLRZPPA	zLinux paging space performance record
ZLINUX_HARDCONF	DRLRZCNF	zLinux hardware configuration record
ZLINUX_SOFTCONF	DRLRZCNF	zLinux software configuration record
ZLINUX_USR_CMD	DRLRZACO	zLinux process/command accounting record
ZLINUX_WTMP_INFO	DRLRZMTP	zLinux connect accounting record
ZLINUX_REC_PI	DRLRLNX1	PI log record reformatted to fixed layout
ZLINUX_REC_DF	DRLRLNX1	DF log record reformatted to fixed layout
ZLINUX_REC_WW	DRLRLNX1	WW log record reformatted to fixed layout
ZLINUX_REC_TO	DRLRLNX1	TO log record reformatted to fixed layout

RACF records

These records come from the RACF Database Unload utility output that contains RACF configuration data.

For a description of these records, refer to *RACF Macros and Interfaces*.

Record name	Member name	Description
RACF_100	DRLRR100	Group basic data
RACF_200	DRLRR200	User basic data
RACF_205	DRLRR205	User connect data
RACF_400	DRLRR400	Data set basic data
RACF_402	DRLRR402	Data set conditional access
RACF_404	DRLRR404	Data set access
RACF_500	DRLRR500	General resource basic data
RACF_505	DRLRR505	General resource access
RACF_507	DRLRR507	General resource conditional access

Tivoli Workload Scheduler for z/OS (OPC) records

These records come from the OPC track log.

For a description of these records, refer to the *Tivoli Workload Scheduler: Diagnosis Guide and Reference*.

Record name	Member name	Description
OPC_03_P	DRLROP03	OPC current plan operation
OPC_03_C	DRLROP03	OPC current plan occurrence
OPC_03_3	DRLROP03	OPC current plan system automation
OPC_04	DRLROP04	OPC current plan job name table
OPC_23	DRLROP23	OPC operation event
OPC_24	DRLROP24	OPC MCP event
OPC_27	DRLROP27	OPC missed feedback
OPC_29	DRLROP29	OPC auto tracked event

VM accounting records

For a description of these records, refer to *z/VM: CP Planning and Administration*.

Record name	Member name	Description
VMACCT_01	DRLRVA01	Virtual machine resource use
VMACCT_02	DRLRVA02	Dedicated devices
VMACCT_03	DRLRVA03	Temporary disk space
VMACCT_04	DRLRVA04	LOGON or AUTOLOG with invalid password
VMACCT_05	DRLRVA05	Successful LINK to protected minidisk
VMACCT_06	DRLRVA06	LINK with invalid password

VM accounting records

Record name	Member name	Description
VMACCT_07	DRLRVA07	Log off from VSCS-controlled device
VMACCT_08	DRLRVA08	Disconnect or log off

VMPRF records

For a description of these records, refer to the *VMPRF User's Guide and Reference*.

Record name	Member name	Description
VMPRF_01	DRLRVM01	VMPRF system data
VMPRF_02	DRLRVM02	VMPRF processor data
VMPRF_11	DRLRVM11	VMPRF configuration data
VMPRF_41	DRLRVM41	VMPRF user data
VMPRF_61	DRLRVM61	VMPRF DASD data

z/VM Performance Toolkit records

For a description of these records, refer to the *z/VM Performance Toolkit* manual.

Record name	Member name	Description
VMPERFT_00	DRLRPT00	System configuration data
VMPERFT_01	DRLRPT01	General system load data
VMPERFT_02	DRLRPT02	Processor load data
VMPERFT_03	DRLRPT03	Logical processor load data (LPAR only)
VMPERFT_04	DRLRPT04	Minidisk cache data
VMPERFT_05	DRLRPT05	CP services activity data
VMPERFT_06	DRLRPT06	Channel busy (HF sampling)
VMPERFT_07	DRLRPT07	Channel measurement facility data
VMPERFT_08	DRLRPT08	Extended channel measurement facility data
VMPERFT_3A	DRLRPT3A	Overall user transaction data
VMPERFT_3C	DRLRPT3C	Shared segment data
VMPERFT_3E	DRLRPT3E	Shared data spaces
VMPERFT_41	DRLRPT41	User resource usage and wait states
VMPERFT_42	DRLRPT42	User class resource usage and wait states (same layout as FC41)
VMPERFT_43	DRLRPT43	System totals for user resource usage and wait states (same layout as FC41)
VMPERFT_44	DRLRPT44	User transactions and response time
VMPERFT_45	DRLRPT45	User class transactions and response time data (same layout as FC44)
VMPERFT_46	DRLRPT46	System totals for user transactions and response time data
VMPERFT_51	DRLRPT51	I/O processor activity data
VMPERFT_55	DRLRPT55	Virtual switch records
VMPERFT_61	DRLRPT61	General DASD data
VMPERFT_65	DRLRPT65	DASD cache data
VMPERFT_68	DRLRPT68	DASD CP owned (system areas)

Record name	Member name	Description
VMPERFT_6F	DRLRPT6F	SCSI device records
VMPERFT_6D	DRLRPT6D	Queued Direct Input Output (QDIO) support
VMPERFT_71	DRLRPT71	DASD SEEKS data
VMPERFT_A2	DRLRPTA2	SFS and BFS server data
VMPERFT_A4	DRLRPTA4	Multitasking users data
VMPERFT_A6	DRLRPTA6	TCP/IP server data
VMPERFT_A7	DRLRPTA7	TCP/IP links data
VMPERFT_A8	DRLRPTA8	Reusable server kernel summary data
VMPERFT_A9	DRLRPTA9	Linux application data

z/VM Performance Toolkit records

Chapter 15. Administration dialog options and commands

This chapter describes actions you can access from primary windows in the Tivoli Decision Support for z/OS administration dialog. These actions include dialog window pull-downs and commands you issue from the command line. These sections describe the actions:

- “Tivoli Decision Support for z/OS dialog options”
- “Tivoli Decision Support for z/OS commands” on page 279

Tivoli Decision Support for z/OS dialog options

These figures list menu bar options for the Tivoli Decision Support for z/OS windows. Under each menu bar option, there is a list of pull-down options available, with references to where the pull-down options are described.

Tivoli Decision Support for z/OS Primary Menu window

Options

Dialog parameters

See “Dialog parameters - variables and fields” on page 58.

Reporting dialog defaults

Refer to the *Guide to Reporting* for more information.

Help

Using help

Refer to the *Guide to Reporting* for more information.

General help

Refer to the *Guide to Reporting* for more information.

Keys help

Refer to the *Guide to Reporting* for more information.

Online books

Refer to the *Guide to Reporting* for more information.

Search information

Refer to the *Guide to Reporting* for more information.

Product information

Displays Tivoli Decision Support for z/OS copyright and release information.

•

Administration window

Other

QMF Refer to the *Guide to Reporting* for more information. If your installation does not use QMF, this item is not selectable.

Tivoli Decision Support for z/OS dialog options

DB2I See “Using available tools to work with the Tivoli Decision Support for z/OS database” on page 111.

ISPF/PDF

Displays the ISPF/PDF primary menu.

Process Tivoli Decision Support for z/OS statements

See “Working with fields in a record definition” on page 157.

Messages

Refer to the *Guide to Reporting* for more information.

Exit Returns to the previous window.

Utilities

Network

Refer to the *Network Performance Feature Installation and Administration* manual .

Generate problem records

See “Administering problem records” on page 121.

System Diagnostics

Refer to the topic "System Diagnostics" in the *Messages and Problem Determination* manual.

TPM Extract

Extracts usage data from Tivoli Decision Support for z/OS data tables which can be imported into Tivoli Performance Modeller.

Search installed objects

Utility for searching installed component objects such as table columns, table comments, records, updates, and reports.

Help

Using help

Refer to the *Guide to Reporting* for more information.

General help

Refer to the c for more information.

Keys help

Refer to the *Guide to Reporting* for more information.

Online books

Refer to the *Guide to Reporting* for more information.

Search information

Refer to the *Guide to Reporting* for more information.

Product information

Displays Tivoli Decision Support for z/OS copyright and release information.

•

Components window

Component

New See “Creating a component” on page 140.

Open component

See "Viewing objects in a component" on page 137.

Install See "Installing a component" on page 126.

Uninstall

See "Uninstalling a component" on page 134.

Delete See "Deleting a component" on page 140.

Print list

See "Printing a list of Tivoli Decision Support for z/OS tables" on page 197 for a description of a similar action, printing a list of tables.

Show user objects

See "Controlling objects that you have modified" on page 136.

Show excluded

See "Controlling objects that you have modified" on page 136.

Exit Saves changes and returns to the previous window.

Space

Tablespaces

See "Installing a component" on page 126.

Indexes

See "Installing a component" on page 126.

Other

QMF Refer to the *Guide to Reporting* for more information. If your installation does not use QMF, this item is not selectable.

DB2I See "Using available tools to work with the Tivoli Decision Support for z/OS database" on page 111.

ISPF/PDF

Displays the ISPF/PDF primary menu.

Process Tivoli Decision Support for z/OS statements

See "Working with fields in a record definition" on page 157.

Messages

Refer to the *Guide to Reporting* for more information.

Help

Using help

Refer to the *Guide to Reporting* for more information.

General help

Refer to the *Guide to Reporting* for more information.

Keys help

Refer to the *Guide to Reporting* for more information.

Online books

Refer to the *Guide to Reporting* for more information.

Search information

Refer to the *Guide to Reporting* for more information.

Tivoli Decision Support for z/OS dialog options

Product information

Displays Tivoli Decision Support for z/OS copyright and release information.

•



Logs window

Log

New See “Creating a log definition” on page 153.

Open log definition

See “Viewing and modifying a log definition” on page 152.

Open record definitions

See “Viewing and modifying a record definition” on page 155.

Open collected log data sets

See “Viewing a list of log data sets collected” on page 144.

Open Log Data Manager

See Chapter 10, “Working with the log data manager option,” on page 205.

Delete See “Deleting a log definition” on page 154.

Save definition

See “Saving a table definition in a data set” on page 197 for a description of a similar action, saving definitions for tables.

Print list

See “Printing a list of Tivoli Decision Support for z/OS tables” on page 197 for a description of a similar action, printing a list of tables.

Exit Saves changes and returns to the previous window.

Utilities

Collect

See “Collecting data from a log into DB2 tables” on page 146.

Display log

See “Displaying the contents of a log” on page 149.

Show log statistics

See “Displaying log statistics” on page 148.

View

All Lists all logs in the Logs window.

Some Restricts the list of logs displayed in the Logs window when you specify selection criteria.

Other

QMF Refer to the *Guide to Reporting* for more information. If your installation does not use QMF, this item is not selectable.

DB2I See “Using available tools to work with the Tivoli Decision Support for z/OS database” on page 111.

ISPF/PDF

Displays the ISPF/PDF primary menu.

Process Tivoli Decision Support for z/OS statements

See “Working with fields in a record definition” on page 157.

Messages

Refer to the *Guide to Reporting* for more information.

Help

Using help

Refer to the *Guide to Reporting* for more information.

General help

Refer to the *Guide to Reporting* for more information.

Keys help

Refer to the *Guide to Reporting* *Guide to Reporting* for more information.

Online books

Refer to the *Guide to Reporting* for more information.

Search information

Refer to the *Guide to Reporting* for more information.

Product information

Displays Tivoli Decision Support for z/OS copyright and release information.

•

Tables window

Table

New See “Creating a table” on page 198.

Open table definition

See “Opening a table to display columns” on page 179.

Open updates

See “Displaying and modifying update definitions of a table” on page 184.

Open purge conditions

See “Displaying and editing the purge condition of a table” on page 190.

Open tablespace

See “Displaying and modifying a table or index space” on page 192.

Delete See “Deleting a table or view” on page 200.

Save definition

See “Saving a table definition in a data set” on page 197.

Print list

See “Printing a list of Tivoli Decision Support for z/OS tables” on page 197.

Exit Saves changes and returns to the previous window.

Maintenance

Tablespace

See “Displaying and modifying a table or index space” on page 192.

Index and indexspace

See “Displaying and modifying a table or index space” on page 192.

Tivoli Decision Support for z/OS dialog options

Utilities

Display

See “Displaying the contents of a table” on page 164.

Show size

See “Showing the size of a table” on page 167.

Import

See “Importing the contents of an IXF file to a table” on page 171. If your installation does not use QMF, this item is not selectable.

Export See “Exporting table data to an IXF file” on page 172. If your installation does not use QMF, this item is not selectable.

Grant See “Administering user access to tables” on page 202.

Revoke

See “Administering user access to tables” on page 202.

Recalculate

See “Recalculating the contents of a table” on page 169.

Purge See “Purging a table” on page 172.

Unload

See “Unloading and loading tables” on page 172.

Load See “Unloading and loading tables” on page 172.

Edit

Add rows

See “Editing the contents of a table” on page 165. If your installation does not use QMF, this item is not selectable.

Change rows

See “Editing the contents of a table” on page 165. If your installation does not use QMF, this item is not selectable.

ISPF editor

See “Editing the contents of a table” on page 165.

View

All See “Listing a subset of tables in the Tables window” on page 198.

Some See “Listing a subset of tables in the Tables window” on page 198.

Other

QMF Refer to the *Guide to Reporting* for more information. If your installation does not use QMF, this item is not selectable.

DB2I See “Using available tools to work with the Tivoli Decision Support for z/OS database” on page 111.

ISPF/PDF

Displays the ISPF/PDF primary menu.

Process Tivoli Decision Support for z/OS statements

See “Working with fields in a record definition” on page 157.

Messages

Refer to the *Guide to Reporting* for more information.

Help

Using help

Refer to the *Guide to Reporting* for more information.

General help

Refer to the *Guide to Reporting* for more information.

Keys help

Refer to the *Guide to Reporting* *Guide to Reporting* for more information.

Online books

Refer to the *Guide to Reporting* for more information.

Search information

Refer to the *Guide to Reporting* for more information.

Product information

Displays Tivoli Decision Support for z/OS copyright and release information.

Tivoli Decision Support for z/OS commands

You can immediately execute an action anywhere in a Tivoli Decision Support for z/OS dialog by typing these commands on the command line (uppercase letters show the abbreviation for the command):

COMPonen (see Note)

Displays the Components window.

DB2I Starts a DATABASE 2 Interactive (DB2I) facility session and displays its primary menu.

DISPLay RECORD record_type (see Note)

Lets you identify a log data set in the Record Selection window from which Tivoli Decision Support for z/OS displays records of the specified type in the Record Data window.

DISPLay report_ID

Displays the specified report from the Reports window.

DISPLay REPort report_ID

Displays the specified report. By default, report IDs are listed in the Tivoli Decision Support for z/OS Report window next to their corresponding report descriptions. You can toggle the display to show either the report IDs or the report types and owners by pressing F11.

If you do not use a prefix for the report ID (*prefix.report_ID*), Tivoli Decision Support for z/OS assumes the report is public. Otherwise, the prefix must be the owner of the private report.

DISPLAY TABLE table_name (see Note)

Displays the specified table.

Tivoli Decision Support for z/OS assumes a prefix that is the value of the Other table prefix field from the Dialog Parameters window:

```
DISPL TAB DRLSYS.DRLTABLES  
DISPL TAB MVS_SYSTEM_H or DISPL TAB DRL.MVS_SYSTEM_H
```

DISPLay table_name (see Note)

Displays the specified table from the Tables window.

DRLESTRA

Displays the Set/Reset Trace Options window.

HELP Displays general help or, if a message appears, help for the message.

INFO Calls BookManager and displays the Topics in Online Books window.

INFO SEarCh

Calls BookManager and displays the BookManager Set Up Search window.

Administration dialog commands

INFO Search **argument**

Calls BookManager and searches for *argument*. If you omit *argument*, this command calls BookManager to display the Set Up Search pop-up.

ISPF Displays the ISPF primary menu.

LOcate **argument**

In a Tivoli Decision Support for z/OS window, locates the first row that starts with *argument* in the column that was last sorted.

LOGS (see Note)

Displays the Logs window.

PDF Displays the ISPF primary menu.

QMF If your installation uses QMF, this command starts QMF and displays either its SQL primary window or its prompted query primary menu.

REPORTS

Starts the reporting dialog.

SOrt **column_name** | **position** **ASC** | **DES**

Sorts a Tivoli Decision Support for z/OS list by the column you specify as *column_name* in either ascending or descending order. (You can also sort by column number by specifying the number of the column instead of the name. The first column after the selection field column on the left is column 1.)

SYStem (see Note)

Displays the System window.

TABlE (see Note)

Displays the Tables window.

Note: This command is not available in end-user mode from the reporting dialog.

Chapter 16. Administration reports

This chapter describes the administration reports that are created when you create or update the Tivoli Decision Support for z/OS system tables. The reports listed in this chapter are the following:

- “PRA001 - Indexspace cross-reference”
- “PRA002 - Actual tablespace allocation” on page 282
- “PRA003 - Table purge condition” on page 284
- “PRA004 - List columns for a requested table with comments” on page 285
- “PRA005 - List all tables with comments” on page 286
- “PRA006 - List User Modified Objects” on page 287

PRA001 - Indexspace cross-reference

The PRA001 report provides a cross-reference between indexspaces and indexes that are present in the Tivoli Decision Support for z/OS environment at the time of running the report.

This report enables you to extract the real name of an index, so that you can locate the index in the administration dialog and adjust its space allocation if required.

The source table for this report is the DRLINDEXES system table.

This information identifies the report:

Report ID

PRA001

Report group

ADMIN

Reports Source

DRLINDEXES

Attributes

INDEX, INDEXSPACE, ADMINISTRATION, DB2,

Variables

INDEXSPACE. Optional. Type the index name associated with a single indexspace, or accept the default setting to obtain a complete cross reference between index and indexspace names for all indexes.

Figure 110 on page 282 shows part of a PRA001 report.

PRA001 - Indexspace cross-reference

INDEXSPACE cross-reference

Indexspace Name	Index Name
DRLLLOGSI	DRLLLOGSIX
DRLCOMPR	DRLCOMP_PART_IX
DRLCOMPO	DRLCOMPONENT_IX
DRLRECOR	DRLRECORDSIX
DRLFIEFD	DRLFIELDSIX
DRLSECTI	DRLSECTIONSIX
DRLRPROC	DRLRPROCINPUTIX
DRLRIMX\$	DRLRECORDPROCSIX
DRLUPDAT	DRLUPDATESIX
DRLUPDCO	DRLUPDCOLSIX
DRLUPDLE	DRLUPDLETSIX
DRLPURGE	DRLPURGEIX
DRLUPDIS	DRLUPDISTRIX
DRLLLOGDA	DRLLLOGDATASETSIX
DRLEXPRI	DRLEXPRIX
DAYROFRW	DAY_OF_WEEK_IX
SPECIALR	SPECIAL_DAY_IX
DRLC1F8M	DRLCOMP_OBJ_IX
DRLREPOR	DRLREPORTS_IX
DRLREPRV	DRLREP_VAR_IX
DRLREPR	DRLREP_ATTR_IX
DRLREPRQ	DRLREP_QRY_IX
DRLREPRC	DRLREP_COL_IX
DRLREPRT	DRLREP_TEXT_IX
DRLGROUP	DRLGROUP_IX
DRLGRPRR	DRLGRP_REP_IX
DRLSEARC	DRLSEARCH_IX
..	
..	
..	

Figure 110. Part of an Indexspace Cross-reference report

The report contains the following information:

INDEXSPACE NAME

The name of the indexspace whose index name has been extracted. This is either the name associated with a single indexspace or the complete cross reference between index and indexspace names for all indexes.

INDEX NAME

The name of the index associated with the indexspace.

For information about:

- The DRLINDEXES system table, see “Views on DB2 and QMF tables” on page 241.
- How to run reports, see “Administering reports” on page 112.
- How to display or modify tables or indexspaces, see “Displaying and modifying a table or index space” on page 192.

PRA002 - Actual tablespace allocation

The PRA002 report shows the actual space allocated to tables. Use the information in this report, together with the information in PRA003, to estimate future space requirements.

The source table for this report is the DRLTABLESPACE system table.

This information identifies the report:

Report ID

PRA002

Report group

ADMIN

Reports Source

DRLTABLESPACE

Attributes

TABLESPACE, SPACE, ADMINISTRATION, DB2,

Variables

TABLESPACE_NAME. Optional. You can select the space allocated to a single tablespace, or accept the default to obtain complete information for all the Tablespace present.

Figure 111 shows part of a PRA002 report.

ACTUAL TABLESPACE SPACE allocation

Tablespace Name	SPACE Allocated
DRLSADSM	1584
DRLSCI08	10080
DRLSCOM	20160
DRLSCS01	1056
DRLSCS02	1056
DRLSCS03	1056
DRLSCS04	1056
DRLSCS05	1056
DRLSCS06	1056
DRLSCS07	1056
DRLSCS08	1056
DRLSCS09	1056
DRLSCS10	1056
DRLSCS11	1056
DRLSCS12	1056
DRLSCS13	1056
DRLSCS14	1056
DRLSCS15	1056
DRLSCS16	1056
DRLSCS17	1056
DRLSCS18	1056
DRLSCS19	1056
..	
..	
..	

Figure 111. Part of an Actual Tablespace Allocation report

The report contains the following information:

Tablespace Name

The name of the tablespace whose space allocation has been extracted.

SPACE Allocated

The SPACE value as reported in the DB2 catalog (SYSIBM.SYSTABLESPACES table). The column SPACE contains data only if the STOSPACE utility has been run.

For information about:

- The DRLTABLESPACE system table, see “Views on DB2 and QMF tables” on page 241.
- How to run reports, see “Administering reports” on page 112.

PRA002 - Actual tablespace allocation

- How to display or modify tables or indexspaces, see “Displaying and modifying a table or index space” on page 192.
- The SYSTABLESPACE table, refer to the *DB2 Universal Database for OS/390 and z/OS: SQL Reference*.

PRA003 - Table purge condition

This report shows a printable list of current purge conditions. It enables you to review purge criteria and decide which adjustments to make without the need to use the online dialog.

The source table is the DRLPURGCOND system table.

This information identifies the report:

Report ID

PRA003

Report group

ADMIN

Reports Source

DRLPURGECOND

Attributes

TABLE, PURGE, ADMINISTRATION, DB2,

Variables

TABLE_NAME. Optional. You can select the purge condition associated with a single table, or accept the default setting to obtain a complete list of current purge conditions.

Figure 112 on page 285 shows part of a PRA003 report.

TABLE PURGE Condition

Table Name	Purge Condition
TCP_GEN_IP_H	DATE < CURRENT DATE - 7 DAYS
TCP_GEN_TCP_H	DATE < CURRENT DATE - 7 DAYS
TCP_API_CALLS_D	DATE < CURRENT DATE - 30 DAYS
TCP_API_CALLS_H	DATE < CURRENT DATE - 7 DAYS
TCP_GEN_UDP_H	DATE < CURRENT DATE - 7 DAYS
TCP_GEN_ICMP_H	DATE < CURRENT DATE - 7 DAYS
TCP_GEN_IP_D	DATE < CURRENT DATE - 30 DAYS
TCP_GEN_TCP_D	DATE < CURRENT DATE - 30 DAYS
TCP_GEN_UDP_D	DATE < CURRENT DATE - 30 DAYS
TCP_GEN_ICMP_D	DATE < CURRENT DATE - 30 DAYS
TCP_GEN_IP_W	DATE < CURRENT DATE - 365 DAYS
TCP_GEN_TCP_W	DATE < CURRENT DATE - 365 DAYS
TCP_GEN_UDP_W	DATE < CURRENT DATE - 365 DAYS
TCP_GEN_ICMP_W	DATE < CURRENT DATE - 365 DAYS
TCP_FTP_CLIENT_T	DATE < CURRENT DATE - 1 DAYS
TCP_FTP_CLIENT_H	DATE < CURRENT DATE - 7 DAYS
TCP_FTP_CLIENT_D	DATE < CURRENT DATE - 30 DAYS
TCP_FTP_CLIENT_W	DATE < CURRENT DATE - 365 DAYS
TCP_FTP_SERVER_T	DATE < CURRENT DATE - 1 DAYS
TCP_FTP_SERVER_H	DATE < CURRENT DATE - 7 DAYS
TCP_FTP_SERVER_D	DATE < CURRENT DATE - 30 DAYS
TCP_FTP_SERVER_W	DATE < CURRENT DATE - 365 DAYS
TCP_TN3270_CLNT_T	DATE < CURRENT DATE - 1 DAYS
TCP_TN3270_CLNT_H	DATE < CURRENT DATE - 7 DAYS
..	
..	
..	

Figure 112. Part of a Table Purge Condition report

The report contains the following information:

TABLE NAME

The name of the table to which the purge condition applies.

PURGE CONDITION

The purge condition that applies to the table.

For information about:

- The DRLPURGCOND system table, see “Views on DB2 and QMF tables” on page 241.
- How to run reports, see “Administering reports” on page 112.
- How to display or edit purge conditions, see “Displaying and editing the purge condition of a table” on page 190.

PRA004 - List columns for a requested table with comments

This report shows the column remarks for the selected table.

This information identifies the report:

Report ID

PRA004

Report group

ADMIN

Reports Source

DRLCOLUMNS

PRA004 - List columns for a requested table with comments

Attributes

COMMENT, PURGE, ADMINISTRATION, DB2, idd:break>TABLE

Variables

Tablename.

List columns for a requested table with comments

```
TABLE: DB2_USER_TRAN_H
KEYS  KEYSEQ  NAME                COLTYPE  LENGTH  REMARKS
-----
K      7        CORRELATION_ID      CHAR     12      Correlat. ID value. From QWHCCV.
K     10        DB2_PLAN             CHAR      8      Plan name. From QWHCLAN.
      0        BP32_DYN_PREFETCH   FLOAT     4      Num. of DYNAMIC PREFETCH requests
      0        BP32_EXPANSIONS     FLOAT     4
```

Tivoli Decision Support for z/OS: PRA004

Figure 113. Example of List columns for a requested table with comment

The report contains the following information:

Keys K Indicates if the column is primary Key in the table.

Keyseq

The column's numeric position within the table's primary key. 0 if it is not part of a primary key.

Name Table column name.

Coltype

The type attribute associated to the column.

Length

Column length.

Remarks

Column comment (if defined for the table column). It is 255 char long.

PRA005 - List all tables with comments

This report lists all the tables with remarks.

This information identifies the report:

Report ID

PRA005

Report group

ADMIN

Reports Source

DRLCOLUMNS

Attributes

COMMENT,PURGE,ADMINISTRATION,DB2, TABLE

Variables

Tablename.

```

List all tables with comments
NAME          REMARKS
-----
CICS_DICTIONARY CICS dictionary records. Used and maintained by the recordproc that
CICS_FIELD      CICS dictionary fields. Used to update the dictionary blocks in
DAY_OF_WEEK     This control table defines the day type to be returned by the
                DAYTYPE
DB2_APPL_DIST_H This table provides hourly statistics on DDF distributed address
                space
    
```

Tivoli Decision Support for z/OS: PRA005

Figure 114. Example of List all tables with comment

The report contains the following information:

Name Table column name.

Coltype

The type attribute associated to the column.

Length

Column length.

Remarks

Table comment. It is 255 characters long.

PRA006 - List User Modified Objects

The PRA006 report provides the list of all the user-modified objects, that is, the objects that have a version value different from 'IBM.xxx'. The source tables for this report are the DRLCOMP_OBJECTS, DRLRECORDS, DRLRECORDPROCS, DRLLOGS, DRLUPDATES, DRLREPOSTS system tables.

This information identifies the report.

Report ID

PRA006

Report group

ADMIN

Reports Source

DRLCOMP_OBJECTS, DRLRECORDS, idd:break>DRLRECORDPROCS, DRLLOGS, DRLUPDATES, DRLREPOSTS

Attributes

USER, CHANGES, OBJECTS, ADMINISTRATION

Variables

COMPONENT. Optional. Type a component name if you want the user-modified objects for a single component. If you do not specify any value, the complete list of user modified objects is displayed for each installed component.

The following is an extract from a PRA006 report:

PRA006 - List User Modified Objects

List User Modified Objects

Component Name	Object Type	Object Name	Member Name	Part Name	Version
ADSM	LOG	SMF	DRLLSMF	-	FLAG
CICSMON	LOG	SMF	DRLLSMF	-	FLAG
	RECORD	SMF_110_1	DRLRS110	-	FLAG
	REPORT	CICSA05	DRLOC107	7 CMF GLOB & ACCT	PN86655
		CICSA07	DRLOC107	7 CMF GLOB & ACCT	PN86655
	UPDATE	CICS_TRAN_USR_H	DRLTCITR	1 CMF BASIC	ALTERED
CICSMOP	LOG	SMF	DRLLSMF	-	FLAG
	RECORD	SMF_110_1	DRLRS110	-	FLAG
CICSSTAP	LOG	SMF	DRLLSMF	-	FLAG
	RECORD	SMF_110_2	DRLR1102	-	PQ03356
		SMF_110_2_02	DRLR1102	-	PQXXXXX
		SMF_110_2_07	DRLR1102	-	PQ03356
		SMF_110_2_08	DRLR1102	-	PQ03356
		SMF_110_2_10	DRLR1102	-	PQ03356
		SMF_110_2_108	DRLR1102	-	PQXXXXX
		SMF_110_2_11	DRLR1102	-	PQXXXXX
		SMF_110_2_12	DRLR1102	-	PQ03356
		SMF_110_2_16	DRLR1102	-	PQ03356
		SMF_110_2_17	DRLR1102	-	PQ03356
		SMF_110_2_18	DRLR1102	-	PQ03356
		SMF_110_2_21	DRLR1102	-	PQ03356
		SMF_110_2_23	DRLR1102	-	PQ03356
		SMF_110_2_24	DRLR1102	-	PQ03356
		SMF_110_2_25	DRLR1102	-	PQ03356
		SMF_110_2_28	DRLR1102	-	PQ03356
		SMF_110_2_30	DRLR1102	-	PQ03356
		SMF_110_2_34	DRLR1102	-	PQ03356
		SMF_110_2_37	DRLR1102	-	PQ28635
		SMF_110_2_39	DRLR1102	-	PQXXXXX
		SMF_110_2_40	DRLR1102	-	PQ28635
		SMF_110_2_45	DRLR1102	-	PQXXXXX
		SMF_110_2_48	DRLR1102	-	PQXXXXX
		SMF_110_2_48_2	DRLR1102	-	PQXXXXX
		SMF_110_2_54	DRLR1102	-	PQXXXXX
		SMF_110_2_61	DRLR1102	-	PQ03356
		SMF_110_2_63	DRLR1102	-	PQ03356
		SMF_110_2_66	DRLR1102	-	PQ03356
		SMF_110_2_76	DRLR1102	-	PQ03356
		SMF_110_2_81	DRLR1102	-	PQ03356
		SMF_110_2_85	DRLR1102	-	PQ03356
		SMF_110_2_87	DRLR1102	-	PQ03356
		SMF_110_2_88	DRLR1102	-	PQ03356
		SMF_110_2_90	DRLR1102	-	PQXXXXX
		SMF_110_2_93	DRLR1102	-	PQ03356
		SMF_110_2_94	DRLR1102	-	PQ03356
		SMF_110_3	DRLR1103	-	PQ03356
	RECPROC	DRL2CIST	DRLR1103	-	ALTERED
	UPDATE	CICS_S_TCPIP_DP	DRLTS3P7	-	PQXXXXX
		CICS_S_TCPIP_TP	DRLTS3P7	-	PQXXXXX

The report contains the following information:

Component Name

Name of the component which the objects belong to.

Object Type

Type of object (Record, Update, Log...).

Object Name

Name of the object.

Member Name

Name of the member in the Tivoli Decision Support for z/OS libraries where the object definition is stored.

Part Name

Subcomponent name, if any.

Version

Version of the object. You modify this field when you change any objects. It indicates whether an object has been modified.

For information about:

- The DRLCOMP_OBJECTS, DRLRECORDS, DRLRECORDPROCS, DRLLOGS, DRLUPDATES, DRLREPOSTS system tables, see “Views on DB2 and QMF tables” on page 241.
- How to run reports, see “Administering reports” on page 112.

PRA006 - List User Modified Objects

Chapter 17. Using the REXX-SQL interface

This chapter contains General-use Programming Interface and Associated Guidance Information.

Tivoli Decision Support for z/OS provides a REXX-SQL interface through the DRL1SQLX module, which supports:

- Loading a DB2 table into an array of REXX variables
- Using SQL EXECUTE IMMEDIATE to execute an argument string that is a valid SQL statement

For more information about DB2 terms and statements mentioned in this chapter, refer to the *DB2 Universal Database for OS/390 and z/OS: SQL Reference*.

Calling the DRL1SQLX module

The module derives its input data from the argument on the CALL instruction and from predefined REXX variables. There are reserved REXX variables that the calling REXX exec defines before calling the module.

If a REXX exec passes an SQL SELECT statement as the argument, DRL1SQLX executes the SELECT and returns table data in an array of REXX variables. The module can return any DB2 data type but graphic strings.

The module return code result, set in the variable RESULT, is available to the calling REXX program.

The syntax for running the DRL1SQLX module is:

```
▶▶—CALL DRL1SQLX—' INIT'—————▶▶
                    |
                    |—sql-statement—|
                    |
                    | ' TERM'          |
```

where:

INIT Establishes a call attachment facility (CAF) connection to DB2 that leaves the connection open until a DRL1SQLX TERM statement is executed. There is not an implied COMMIT until the DRL1SQLX TERM statement.

If the REXX program passes INIT as the argument for the CALL DRL1SQLX statement, the connection remains open for each SQL statement call. The connection does not terminate until a CALL DRL1SQLX TERM statement closes it.

If the REXX program does not pass INIT as the argument for the CALL DRL1SQLX statement, the connection is opened at the beginning of each CALL DRL1SQLX $sql_statement$ and closed at its conclusion, which makes SQL ROLLBACK impossible.

If you are making more than three calls to DRL1SQLX, it is more efficient to use the CALL DRL1SQLX INIT statement first.

$sql_statement$

An SQL SELECT or another SQL statement that can be executed with an

Calling the DRL1SQLX module

EXECUTE IMMEDIATE statement. DRL1SQLX appends the SQL statement to SQL EXECUTE IMMEDIATE and executes it.
TERM Terminates an existing connection to DB2 and performs an implied COMMIT.

Input REXX variables

The calling program can define these variables before calling DRL1SQLX:

DB2SUBS

The DB2 subsystem that DRL1SQLX addresses.

There is no default for this variable; it must be defined.

DB2PLAN

The name of the DB2 application plan. This variable should be coded only if the installation changed the default plan name DRLPLAN when the Tivoli Decision Support for z/OS bind job was run.

SQLSTEM

The stem of the REXX array that DRL1SQLX uses to return table values when the argument is an SQL SELECT statement.

The stem has an initial value of SQLDATA.

SQLMAX

The maximum number of rows to fetch when the argument is an SQL SELECT statement.

SQLMAX has an default value of 5000. Pick an SQLMAX limit that protects you from runaway queries. The maximum supported value is 99999999.

Output REXX variables

DRL1SQLX always sets these variables:

RESULT

The DRL1SQLX return code.

When the argument is an SQL SELECT, DRL1SQLX sets RESULT to 4 if the number of rows in the table is greater than the value of SQLMAX. It issues a message, DRL1007W, to warn you of the condition but completes the select, returning the number of rows specified in SQLMAX.

DRL1SQLX sets these return codes in RESULT:

- 0 Successful execution.
- 4 SQLCODE > 0, SQLMAX invalid or the SQLMAX limit was reached. The error message is in SQLMSG.
- 8 SQLCODE < 0 indicates an SQL error. The error message is in SQLMSG.
- 12 An error that is not an SQL error. The error message is in SQLMSG.
- 16 There was either insufficient REXX storage or a REXX variable that could not be set. The error appears in SQLMSG, if possible.
- 20 The REXX communication routine IRXEXCOM could not be loaded. There is no indication of the error in SQLMSG.

SQLCODE

The SQL return code.

This value is positive when there is an SQL warning and negative when there is an SQL error. It is returned in combination with a RESULT of 4 or 8, exclusively.

SQLMSG.0

The number of different message values returned when RESULT > 0

SQLMSG.1

The value of the first message returned when RESULT > 0

Up to 5 messages can be returned.

SQLMSG.n

The value of the last message returned when RESULT > 0

The value of *n* is the value of SQLMSG.0.

These variables are set by DRL1SQLX after a successful execution of an SQL SELECT statement. For each variable below, *sqlstem* is the value of the SQLSTEM input variable, *y* is the column number, and *z* is the row number:

sqlstem.NAME.0

The number of selected columns.

sqlstem.NAME.y

The names of the selected columns.

The column name of an expression is blank. Each value of *y* is a whole number from 1 through *sqlstem.NAME.0*.

sqlstem.LENGTH.y

The maximum length of the value of the selected columns.

A column name can be longer than the value. Each value of *y* is a whole number from 1 through *sqlstem.NAME.0*.

sqlstem.TYPE.y

The data types of the selected columns.

Each type is copied from the SQLTYPE field in the SQL descriptor area (SQLDA) and is a number ranging from 384 to 501. Each value of *y* is a whole number from 1 through *sqlstem.NAME.0*.

sqlstem.0

The number of rows in the result table.

sqlstem.y.z

The value of the column.

Each value of *y* is a whole number from 1 through *sqlstem.NAME.0*.

Each value of *z* is a whole number from 1 through *sqlstem.0*.

Reserved REXX variable

DRL1SQLX always sets the variable SQLHANDLE on the INIT statement. It must not be reset except by the TERM statement, which must be able to read the value set by the last INIT statement.

SQLHANDLE contains the handle returned when DRL1SQLX connects to DB2 with the INIT statement.

Calling the DRL1SQLX module

REXX example of calling DRL1SQLX

```
/**REXX*****  
/* Execute an SQL SELECT statement and display output */  
/*****  
  
sqlstmt = "SELECT *",  
          "FROM DRL.MVS_SYSTEM_H",  
          "WHERE DATE = '2000-05-02'"  
  
db2subs = 'DB2T'           /* subsystem name      */  
sqlstem = 'RES'           /* name of stem        */  
sqlmax = 100              /* limit on nbr of rows */  
  
Call DRL1SQLX sqlstmt     /* execute SQL statement */  
  
Say 'DRL1SQLX return code:' result  
Say 'SQL return code SQLCODE:' sqlcode  
  
If sqlmsg.0 > 0 Then  
  Do n = 1 To sqlmsg.0    /* up to 5 error msgs */  
    Say sqlmsg.n  
  End  
  
If res.name.0 > 0 Then   /* number of columns */  
  
  /*****  
  /* Display column names and values for all rows */  
  /*****  
  If res.0 > 0 Then      /* number of rows */  
    Do z = 1 To res.0  
      Say ' '  
      Say 'Following values were returned for row 'z':'  
      Do y = 1 To res.name.0  
        Say res.name.y': 'res.y.z  
      End  
    End  
  Else  
    Say 'No rows were returned'  
Exit
```

Figure 115. Example of REXX-SQL interface call

Chapter 18. Using the IBM DB2 Analytics Accelerator

The IBM DB2 Analytics Accelerator is a high-performance appliance that integrates business insights into operational processes.

Tivoli Decision Support for z/OS includes Analytics Components that are designed to support the IBM DB2 Analytics Accelerator. These components are based on existing non-Analytics components that are modified to allow for the following functions:

- Store data directly to an IBM DB2 Analytics Accelerator removing the need to store data on DB2 for z/OS.
- Allow for more detailed timestamp level records to be stored.
- Allow for more CPU work to move from z/OS to the IBM DB2 Analytics Accelerator appliance.
- Report to make use of the high query speeds of the IBM DB2 Analytics Accelerator.

The System Data Engine component of the IBM Common Data Provider for z Systems[®] is used to convert SMF log data into data sets that contain the IBM Tivoli Decision Support for z/OS Analytics components tables in DB2 internal format. The IBM DB2 Analytics Accelerator Loader for z/OS is then used to load the DB2 internal format data sets directly into the IBM DB2 Analytics Accelerator.

The Analytics components comprise the following items:

- Analytics - z/OS Performance
- Analytics - DB2
- Analytics - KPM CICS
- Analytics - KPM DB2
- Analytics - KPM z/OS

Relationship of Analytics Components to non-Analytics Components

The Analytics components are based on the following existing non-Analytics components:

Table 8. Relationship of Analytics components to non-Analytics components

Analytics	Non-Analytics
Analytics - DB2	DB2
Analytics - KPM CICS	Key Performance Metrics - CICS
Analytics - KPM DB2	Key Performance Metrics - DB2
Analytics - KPM z/OS	Key Performance Metrics – z/OS
Analytics - z/OS Performance	z/OS Performance Management (MVSPM)

The Analytics components include Lookup tables that must be customized as per their equivalent Lookup tables in the non-Analytics components:

Table 9. Relationship of Analytics Lookup table to non-Analytics Lookup table

Member name	Analytics Lookup table	non-Analytics Lookup table
DRLTA2AP	A_DB2_APPLICATION	DB2_APPLICATION
DRLTA2AC	A_DB2_ACCUMAC	DB2_ACCUMAC
DRLTALUG	A_USER_GROUP	USER_GROUP
DRLTALKP	A_KPM_THRESHOLDS_L	KPM_THRESHOLDS
DRLTALW2	A_WORKLOAD2_L	MVS_WORKLOAD2_TYPE
DRLTALDA	A_DEVICE_ADDR_L	MVSPM_DEVICE_ADDR
DRLTALUT	A_UNIT_TYPE_L	MVSPM_UNIT_TYPE
DRLTALMI	A_MIPS_L	MVS_MIPS_T
DRLTALSP	A_SYSPLEX_L	MVS_SYSPLEX
DRLTALWL	A_WORKLOAD_L	MVS_WORKLOAD_TYPE
DRLTALTR	A_TIME_RES_L	MVSPM_TIME_RES

The following table lists all the reports per Analytics component, and their equivalent non-Analytics component reports.

Table 10. Relationship of Analytics component report to non-Analytics component report

Report name	Analytics Report ID	non-Analytics Report ID
DB2 Buffer Pool Exceptions	ADB219	DB219
DB2 Buffer Pool Statistics, Detail	ADB215	DB215
DB2 Buffer Pool Statistics, Overview	ADB216	DB216
DB2 DBRMs Class 7,8 Times, Overview	ADB222	DB222
DB2 General Measure by Profile, Overview	ADB212	DB212
DB2 IDAA Statistics by Transaction, Detail	ADB244	DB244
DB2 Packages Class 7,8 Times, Overview	ADB221	DB221
DB2 Transaction Statistics, Detail	ADB204	DB204
DB2 Transaction Statistics, Overview	ADB205	DB205

The following table lists all the tables per Analytics component, and their equivalent non-Analytics component tables.

Table 11. Relationship of Analytics component table to non-Analytics component table

Component	Type	Analytics component table	Equivalent to non-Analytics component table
Analytics - z/OS Performance	Table	A_PM_CF_I	MVSPM_CF_H
		A_PM_CF_LINK_I	MVSPM_CF_LINK_H
		A_PM_CF_PROC_I	MVSPM_CF_PROC_H
		A_PM_CF_REQ_I	MVSPM_CF_REQUEST_H
		A_PM_CF_CF_I	MVSPM_CF_TO_CF_H
		A_PM_XCF_MEMBER_I	MVSPM_XCF_MEMBER_H
		A_PM_XCF_PATH_I	MVSPM_XCF_PATH_H
		A_PM_XCF_SYS_I	MVSPM_XCF_SYS_H
		A_PM_OMVS_BUF_I	MVSPM_OMVS_BUF_H
		A_PM_OMVS_FILE_I	MVSPM_OMVS_FILE_H
		A_PM_OMVS_GHFS_I	MVSPM_OMVS_GHFS_H
		A_PM_OMVS_HFS_I	MVSPM_OMVS_HFS_H
		A_PM_OMVS_KERN_I	MVSPM_OMVS_KERN_H
		A_PM_OMVS_MOUNT_I	MVSPM_OMVS_MOUNT_H
		A_PM_SYS_CLUST_I	MVSPM_CLUSTER_H
		A_PM_SYS_CPU_I	MVSPM_CPU_H
		A_PM_SYS_CPUMT_I	MVSPM_CPUMT_H
		A_PM_SYS_ENQ_I	MVSPM_ENQUEUE_H
		A_PM_SYS_LPAR_I	MVSPM_LPAR_H
		A_PM_SYS_SYS_I	MVSPM_SYSTEM_H
		A_PM_SYS_PROD_I	MVSPM_PROD_T
		A_PM_SYS_PRDINT_I	MVSPM_PROD_INT_T
		A_PM_SYS_MSU_I	MVSPM_LPAR_MSU_T
		A_PM_WL_GOAL_I	MVSPM_GOAL_ACT_H
		A_PM_WL_SERVED_I	MVSPM_WLM_SERVED_H
		A_PM_WL_STATE_I	MVSPM_WLM_STATE_H
		A_PM_WL_WKLD_I	MVSPM_WORKLOAD_H
		A_PM_WL_WKLD2_I	MVSPM_WORKLOAD2_H
		A_PM_IO_DATASET_I	MVSPM_DATASET_H
		A_PM_IO_VOLUME_I	MVSPM_VOLUME_H
A_PM_IO_LCU_I	MVSPM_LCU_IO_H		

Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

Component	Type	Analytics component table	Equivalent to non-Analytics component table
Analytics - z/OS Performance	Table	A_PM_GS_BMF_I	MVSPM_BMF_H
		A_PM_GS_CACHE_I	MVSPM_CACHE_H
		A_PM_GS_PAGEDS_I	MVSPM_PAGE_DS_H
		A_PM_GS_PAGING_I	MVSPM_PAGING_H
		A_PM_GS_STORAGE_I	MVSPM_STORAGE_H
		A_PM_GS_STORCLS_I	MVSPM_STORCLASS_H
		A_PM_GS_SWAP_I	MVSPM_SWAP_H
		A_PM_GS_CACHES_I	MVSPM_CACHE_ESS_H
		A_PM_VS_VLF_I	MVSPM_VLF_H
		A_PM_VS_CSASQA_I	MVSPM_VS_CSASQA_H
		A_PM_VS_PRIVATE_I	MVSPM_VS_PRIVATE_H
		A_PM_VS_SUBPOOL_I	MVSPM_VS_SUBPOOL_H
		A_PM_DEV_CHAN_I	MVSPM_CHANNEL_H
		A_PM_DEV_HSCHAN_I	MVSPM_HS_CHAN_H
		A_PM_DEV_AP_I	MVSPM_DEVICE_AP_H
		A_PM_DEV_DEVICE_I	MVSPM_DEVICE_H
		A_PM_DEV_FICON_I	MVSPM_FICON_H
		A_PM_DEV_RAID_I	MVSPM_RAID_RANK_H
		A_PM_DEV_ESSLNK_I	MVSPM_ESSLINK_H
		A_PM_DEV_ESSEXT_I	MVSPM_ESS_EXTENT_H
		A_PM_DEV_ESSRNK_I	MVSPM_ESS_RANK_H
		A_PM_DEV_PCIE_I	MVSPM_PCIE_H
		A_PM_Cryp_PCI_I	MVSPM_CRYPTOPCI_H
		A_PM_Cryp_CCF_I	MVSPM_CRYPTOPCF_H
A_PM_APP_APPL_I	MVSPM_APPL_H		

Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

Component	Type	Analytics component table	Equivalent to non-Analytics component table
Analytics - DB2	Table	A_DB2_SYS_PARM_I	DB2_SYS_PARAMETER
		A_DB2_DB_I	DB2_DATABASE_T
		A_DB2_DB_BIND_I	DB2_DATABASE_T
		A_DB2_DB_QIST_I	DB2_DATABASE_T
		A_DB2_DB_SYS_I	DB2_SYSTEM_T
		A_DB2_BP_I	DB2_BUFFER_POOL_T
		A_DB2_USERTRAN_I	DB2_USER_TRAN_H
		A_DB2_UT_BP_I	DB2_USER_TRAN_H
		A_DB2_UT_SACC_I	DB2_USER_TRAN_H
		A_DB2_UT_IDAA_I	DB2_USER_TRAN_H
		A_DB2_IDAA_STAT_I	DB2_IDAA_STAT_H
		A_DB2_IDAA_ACC_I	DB2_IDAA_ACC_H
		A_DB2_IDAA_ST_A_I	DB2_IDAA_STAT_A_H
		A_DB2_IDAA_ST_S_I	DB2_IDAA_STAT_S_H
		A_DB2_PACK_I	DB2_PACKAGE_H
		A_DB2_SHR_BP_I	DB2_BP_SHARING_T
		A_DB2_SHR_BPAT_I	DB2_BPATTR_SHR_T
		A_DB2_SHR_LOCK_I	DB2_LOCK_SHARING_T
		A_DB2_SHR_INIT_I	DB2_SHARING_INIT
		A_DB2_SHR_TRAN_I	DB2_US_TRAN_SHAR_H
	A_DB2_DDF_I	DB2_USER_DIST_H	
A_DB2_SYSTEM_I	DB2_SYSTEM_DIST_T		
A_DB2_STORAGE_I	DB2_STORAGE_T		
View	A_DB2_TRAN_IV	DB2_TRANSACTION_D	
	A_DB2_DATABASE_IV	DB2_DATABASE_T	

IBM DB2 Analytics Accelerator

Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

Component	Type	Analytics component table	Equivalent to non-Analytics component table
Analytics – KPM DB2	Table	A_KD_UT_I	KPM_DB2_USERTRAN_H
		A_KD_UT_BP_I	KPM_DB2_USERTRAN_H
		A_KD_EU_I	KPM_DB2_ENDUSER_H
		A_KD_EU_BP_I	KPM_DB2_ENDUSER_H
		A_KD_PACKAGE_I	KPM_DB2_PACKAGE_H
		A_KD_SYS_IO_I	KPM_DB2_SYSTEM_T
		A_KD_SYS_TCBSRB_I	KPM_DB2_SYSTEM_T
		A_KD_SYS_LATCH_I	KPM_DB2_LATCH_T
		A_KD_SYS_BP_I	KPM_DB2_BP_T
		A_KD_SYS_BP_SHR_I	KPM_DB2_BP_SHR_T
		A_KD_SYS_ST_DBM_I	KPM_DB2_STORAGE_T
		A_KD_SYS_ST_DST_I	KPM_DB2_STORAGE_T
		A_KD_SYS_ST_COM_I	KPM_DB2_STORAGE_T
		A_DB_SYS_DB_WF_I	KPM_DB2_DATABASE_T
		A_DB_SYS_DB_EDM_I	KPM_DB2_DATABASE_T
A_DB_SYS_DB_SET_I	KPM_DB2_DATABASE_T		
A_DB_SYS_DB_LOCK_I	KPM_DB2_LOCK_T		
Analytics - KPM CICS	Table	A_KC_MON_TRAN_I	KPMC_MON_TRAN_H

Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

Component	Type	Analytics component table	Equivalent to non-Analytics component table
Analytics - KPM z/OS	Table	A_KPM_EXCEPTION_I	KPM_EXCEPTION_T
		A_KZ_JOB_INT_I	KPMZ_JOB_INT_T
		A_KZ_JOB_STEP_I	KPMZ_JOB_STEP_T
		A_KZ_LPAR_I	KPMZ_LPAR_T
		A_KZ_STORAGE_I	KPMZ_STORAGE_T
		A_KZ_WORKLOAD_I	KPMZ_WORKLOAD_T
		A_KZ_CHANNEL_I	KPMZ_CHANNEL_T
		A_KZ_CF_I	KPMZ_CF_T
		A_KZ_CF_STRUC_I	KPMZ_CF_STRUCTR_T
		A_KZ_CPUMF_I	KPMZ_CPUMF_T
		A_KZ_CPUMF1_I	KPMZ_CPUMF1_T
		A_KZ_CPUMF_PT_I	KPMZ_CPUMF_PT_T
		A_KZ_CPUMF1_PT_I	KPMZ_CPUMF1_PT_T
		A_KZ_SRM_WKLD_I	KPMZ_SRM_WKLD_T

There are cases where multiple tables from an Analytics component are combined into a single view. In these cases, the resulting view matches an existing table from a Tivoli Decision Support for z/OS non-Analytics component. See the following table for views in the Analytics components that are based on multiple tables from non-Analytics components.

Table 12. Relationship of Analytics component tables used in view to non-Analytics component tables used in view

Component	View	Analytics component tables used in view	Equivalent to non-Analytics component table
Analytics - DB2	A_DB2_USERTRAN_IV	A_DB2_USERTRAN_I	DB2_USER_TRAN_H
		A_DB2_UT_BP_I	
		A_DB2_UT_SACC_I	
		A_DB2_UT_IDAA_I	
Analytics - DB2	A_DB2_DATABASE_IV	A_DB2_DB_I	DB2_DATABASE_T
		A_DB2_DB_BIND_I	
		A_DB2_DB_QIST_I	
Analytics - KPM DB2	A_KD_USERTRAN_IV	A_KD_UT_I	KPM_DB2_USERTRAN_H
		A_KD_UT_BP_I	

IBM DB2 Analytics Accelerator

Table 12. Relationship of Analytics component tables used in view to non- Analytics component tables used in view (continued)

Component	View	Analytics component tables used in view	Equivalent to non-Analytics component table
Analytics - KPM DB2	A_KD_ENDUSER_IV	A_KD_EU_I A_KD_EU_BP_I	KPM_DB2_ENDUSER_H
Analytics - KPM DB2	A_KD_SYSTEM_IV	A_KD_SYS_IO_I A_KD_SYS_TCBSRB_I	KPM_DB2_SYSTEM_T
Analytics - KPM DB2	A_KD_STORAGE_IV	A_KD_SYS_ST_DBM_I A_KD_SYS_ST_DST_I A_KD_SYS_ST_COM_I	KPM_DB2_STORAGE_T
Analytics - KPM DB2	A_KD_DATABASE_IV	A_DB_SYS_DB_WF_I A_DB_SYS_DB_EDM_I A_DB_SYS_DB_SET_I	KPM_DB2_DATABASE_T

Configuring Analytics Components for use with IBM DB2 Analytics Accelerator

About this task

You can complete the following steps that are required for IBM Tivoli Decision Support for z/OS to use the IBM DB2 Analytics Accelerator to contain the data for the tables of the Analytics components. Tables that are created on an IBM DB2 Analytics Accelerator (IDAA) can be loaded without loading data into DB2, which requires the following items:

- The System Data Engine (SDE) component of the IBM Common Data Provider for z Systems to collect the SMF data instead of using TDSz Collect. The PTFs for APARs OA52196 and OA52200 must be applied.
- The DB2 Analytics Accelerator Loader for z/OS V2.1 that uses IDAA-Only load mode to load the data that is created by the SDE into the IDAA.

The Analytics components comprise the following items:

- Analytics - z/OS Performance
- Analytics - DB2
- Analytics - KPM CICS
- Analytics - KPM DB2
- Analytics - KPM z/OS

The Analytics components allow for tables to be created as either kind of the following tables:

- DB2 for z/OS tables
- IBM DB2 Analytics Accelerator Accelerator-shadow tables
- IBM DB2 Analytics Accelerator Accelerator-only tables

Procedure

1. Ensure to apply the PTFs for APAR PI70968 to the Tivoli Decision Support for z/OS system.
2. Bind the DB2 plan that is used by Tivoli Decision Support for z/OS by specifying the BIND option QUERYACCELERATION(ELIGIBLE) or QUERYACCELERATION(ENABLE). For example, assuming the default plan name to be DRLPLAN, the BIND PACKAGE to set ELIGIBLE for the query acceleration register is as follows:

```
//SYSTSIN DD *
  DSN SYSTEM(DSN)
    BIND PACKAGE(DRLPLAN) OWNER(authid) MEMBER(DRLPSQLX) -
      ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC) -
      QUERYACCELERATION(ELIGIBLE)
    BIND PLAN(DRLPLAN) OWNER(authid) PKLIST(*.DRLPLAN.*) -
      ACTION(REPLACE) RETAIN
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIAXx) -
  LIB('xxxx.RUNLIB.LOAD')
  END
```

For more information about the sample instructions to BIND with QUERYACCELERATION specified, see *SDRLCNTL(DRLJDBIN)*.

3. Modify the DRLFPROF dataset to reflect the settings to apply when installing Analytics components. DRLFPROF is the Tivoli Decision Support for z/OS dataset that contains user modified parameters. The following parameters in DRLFPROF provide support for the IBM DB2 Analytics Accelerator:

def_useaot = "YES" | "NO"

"YES": Tables are created as Accelerator Only Tables.

"NO": Tables are created in DB2 and are suitable for use either as DB2 tables or as IDAA_ONLY tables. The default value is "NO".

def_accelerator = "xxxxxxx"

"xxxxxxx": The name of the Accelerator where the tables reside. Required only if using Accelerator Only Tables.

def_timeint = "H" | "S" | "T"

"H": The timestamp for records is rounded to hourly intervals that is similar to non-Analytics tables with a suffix of "_H" in other components.

"S": The timestamp for records is rounded to intervals of a second that is similar to non-Analytics tables with time field instead of timestamp in other components.

"T": The timestamp for tables is the actual timestamp in the SMF log record that is similar to non-Analytics tables with suffix "_T". The default value is "T".

- 4.

Important: This step is required only if you use Tivoli Decision Support for z/OS to collect and populate the component tables on DB2 for z/OS, or if you use Tivoli Decision Support for z/OS reporting. If you only collect data into the IBM DB2 Analytics Accelerator and does not have the data reside on DB2 for z/OS, configure the lookup tables in the Common Data Provider for z Systems. See the information about collecting data for direct load to the Accelerator in the IBM Common Data Provider for z Systems V1.1.0 User's Guide (SC27-4624-01).

Customize each lookup table in the Analytics components as per the existing

IBM DB2 Analytics Accelerator

Tivoli Decision Support for z/OS non-Analytics lookup tables. For example, insert the same rows that are currently in DB2_APPLICATION into A_DB2_APPLICATION.

5. Install the desired Analytics component(s).
6. Add tables to the Accelerator.

If Tivoli Decision Support for z/OS uses Accelerator Only Tables (AOTs), i.e. the DRLFPROF setting for *def_useaot* is "YES", DB2 creates the tables on the IBM DB2 Analytics Accelerator when the Analytics components are being installed.

If Tivoli Decision Support for z/OS doesn't use AOTs, the tables need to be added to the IBM DB2 Analytics Accelerator. Tables can be added by using the Data Studio Eclipse application, or by using stored procedures. To use stored procedures to add the tables to an IBM DB2 Analytics Accelerator, modify and submit the SDRLCNTL members in the following table:

Table 13. Relationship of SDRLCNTL member name to Analytics component

SDRLCNTL member name	Component
DRLJA2DA	Analytics - DB2
DRLJAPMA	Analytics - z/OS Performance
DRLJAKCA	Analytics - KPM CICS
DRLJAKDA	Analytics - KPM DB2
DRLJAKZA	Analytics - KPM z/OS

7. Load data into lookup tables on the Accelerator.

If Tivoli Decision Support for z/OS uses Accelerator Only Tables (AOTs), the lookup tables are populated on the IBM DB2 Analytics Accelerator when the Analytics components are being installed.

If Tivoli Decision Support for z/OS doesn't use AOTs, the contents of the lookup tables need to be loaded into the IBM DB2 Analytics Accelerator. Modify and submit the SDRLCNTL members in the following table to move the contents of the lookup tables into the Accelerator

Note: Not all components have lookup tables.

Table 14. Relationship of SDRLCNTL member name to Analytics component

SDRLCNTL member name	Component
DRLJAPMK	Analytics - z/OS Performance
DRLJAKDK	Analytics - KPM DB2
DRLJAKZK	Analytics - KPM z/OS

Collecting data for direct load to the Accelerator

To collect data for direct load to tables on an IBM DB2 Analytics Accelerator, the following items are required:

- The System Data Engine (SDE) component of the IBM Common Data Provider for z Systems to collect the SMF data instead of using TDSz Collect. The PTFs for APARs **OA52196** and **OA52200** must be applied.

- The DB2 Analytics Accelerator Loader for z/OS V2.1 by using IDAA-Only load mode to load the data that is created by the SDE into the IDAA.

See the information about collecting data for direct load to the Accelerator in the IBM Common Data Provider for z Systems V1.1.0 User's Guide (SC27-4624- 01).

After the data has been collected, it can be loaded direct to the IBM DB2 Analytics Accelerator.

Loading data into the Accelerator

About this task

The DB2 Analytics Accelerator Loader for z/OS V2.1 (Loader) is used to load the DB2 internal format data sets, which are created by the System Data Engine (SDE), directly into the DB2 Analytics Accelerator (Accelerator) without the data residing in DB2 for z/OS.

The Loader is invoked via the DB2 LOAD utility with the following amendments:

- A **DD** statement that indicates the Loader is to intercept the DB2 LOAD utility:
//HLODUMMY DD DUMMY
- A statement that tells the loader to load data into the Accelerator. This statement indicates the data is only to reside on the IDAA_ONLY Accelerator, the name of the Accelerator, the schema (e.g. DRLxx) and the table name:

```
//SYSIN DD *
LOAD DATA RESUME YES LOG NO INDDN input_data_set_ddname
IDAA_ONLY ON accelerator-name
INTO TABLE DRLxx.table-name FORMAT INTERNAL;
```

Procedure

1. To load the data that is created by the System Data Engine, modify and submit the **SDRLCNTL** members in the following table based on the installed components:

Table 15. Relationship of SDRLCNTL member name to Analytics component

SDRLCNTL member name	Component
DRLJA2DD	Analytics - DB2
DRLJAPMD	Analytics - z/OS Performance
DRLJAKCD	Analytics - KPM CICS
DRLJAKDD	Analytics - KPM DB2
DRLJAKZD	Analytics - KPM z/OS

2. Enable acceleration of tables after first data load.

If Tivoli Decision Support for z/OS uses Accelerator Only Tables (AOTs), i.e. the DRLFPROF setting for *def_useaot* is "YES", you don't need to enable the tables on the IBM DB2 Analytics Accelerator.

If Tivoli Decision Support for z/OS doesn't use AOTs and the DB2 LOAD is the first load for an **IDAA_ONLY** Accelerator table, after the load has been completed, the table must be enabled for acceleration in the Accelerator. Tables can be enabled by using the Data Studio Eclipse application, or by using stored

IBM DB2 Analytics Accelerator

procedures. To use stored procedures to enable the tables, modify and submit the SDRLCNTL members in the following table:

Table 16. Relationship of SDRLCNTL member name to Analytics component

SDRLCNTL member name	Component
DRLJA2DE	Analytics - DB2
DRLJAPME	Analytics - z/OS Performance
DRLJAKCE	Analytics - KPM CICS
DRLJAKDE	Analytics - KPM DB2
DRLJAKZE	Analytics - KPM z/OS

Uninstalling components used with an IBM DB2 Analytics Accelerator

About this task

To uninstall Analytics components that have been configured for use with an IBM DB2 Analytics Accelerator, perform the following steps:

Procedure

1. Remove tables from the Accelerator.

If Tivoli Decision Support for z/OS uses Accelerator Only Tables (AOTs), i.e. the DRLFPROF setting for *def_useaot* is "YES", you don't need to remove tables on the IBM DB2 Analytics Accelerator because the next step will automatically remove them.

If Tivoli Decision Support for z/OS doesn't use AOTs, the tables must be removed from the Accelerator prior to uninstalling the component. Modify and submit the SDRLCNTL members in the following table based on the components to be uninstalled. Modify and submit the SDRLCNTL members in the following table according to the components to be uninstalled.

Table 17. Relationship of SDRLCNTL member name to Analytics component

SDRLCNTL member name	Component
DRLJA2DR	Analytics - DB2
DRLJAPMR	Analytics - z/OS Performance
DRLJAKCR	Analytics - KPM CICS
DRLJAKDR	Analytics - KPM DB2
DRLJAKZR	Analytics - KPM z/OS

2. Uninstall the Analytics component(s) by using Tivoli Decision Support for z/OS menus.

Part 5. Appendixes

Appendix A. Accessibility

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

Accessibility features

These are the major accessibility features you can use with Tivoli Decision Support for z/OS when accessing it via the *IBM Personal Communications* terminal emulator:

- You can operate all features using the keyboard instead of the mouse.
- You can read text through interaction with assistive technology.
- You can use system settings for font, size, and color for all user interface controls.
- You can magnify what is displayed on your screen.

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

IBM and accessibility

See the IBM Accessibility Center web site at <http://www-03.ibm.com/able/> for more information about the commitment that IBM has to accessibility.

Appendix B. Support information

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- “Searching knowledge bases”
- “Obtaining fixes”
- “Receiving weekly support updates” on page 312
- “Contacting IBM Software Support” on page 312

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching the information center

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

Searching the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the **Web search** topic in your information center. In the navigation frame, click **Troubleshooting and support ► Searching knowledge bases** and select **Web search**. From this topic, you can search a variety of resources, including the following:

- IBM technotes
- IBM downloads
- IBM developerWorks®
- Forums and newsgroups
- Google

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

1. Go to the IBM Software Support Web site at <http://www.ibm.com/software/support/>.
2. Click **Downloads and drivers** in the **Support topics** section.
3. Select the **Software** category.
4. Select a product in the **Sub-category** list.
5. In the **Find downloads and drivers by product** section, select one software category from the **Category** list.
6. Select one product from the **Sub-category** list.

7. Type more search terms in the **Search within results** if you want to refine your search.
8. Click **Search**.
9. From the list of downloads returned by your search, click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at <http://www-304.ibm.com/support/customer/sas/f/handbook/home.html>.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support Web site at <http://www.ibm.com/support/us/>.
2. Click **My support** in the upper right corner of the page.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. Click **Edit profile**.
5. In the **Products** list, select **Software**. A second list is displayed.
6. In the second list, select a product segment, for example, **Application servers**. A third list is displayed.
7. In the third list, select a product sub-segment, for example, **Distributed Application & Web Servers**. A list of applicable products is displayed.
8. Select the products for which you want to receive updates, for example, **IBM HTTP Server** and **WebSphere® Application Server**.
9. Click **Add products**.
10. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
11. Select **Please send these documents by weekly email**.
12. Update your e-mail address as needed.
13. In the **Documents** list, select **Software**.
14. Select the types of documents that you want to receive information about.
15. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

Online

Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

By phone

Call 1-800-IBM-4You (1-800-426-4968).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

Online

Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home and click **How to Enroll**.

By phone

For the phone number to call in your country, go to the IBM Software Support Web site at <http://techsupport.services.ibm.com/guides/contacts.html> and click the name of your geographic region.

- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at <https://techsupport.services.ibm.com/ssr/login>.
- For customers with IBMLink, CATIA, Linux, S/390®, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at <http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006>.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at <http://www.ibm.com/servers/eserver/techsupport.html>.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook on the Web* at <http://techsupport.services.ibm.com/guides/contacts.html> and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. “Determining the business impact”
2. “Describing problems and gathering information” on page 314
3. “Submitting problems” on page 314

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1

The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2

The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3

The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4

The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Submit and track problems** on the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>. Type your information into the appropriate problem submission form.

By phone

For the phone number to call in your country, go to the contacts page of the *IBM Software Support Handbook* at <http://techsupport.services.ibm.com/guides/contacts.html> and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might 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 changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Notices

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 Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. 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 IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Glossary

A

administration

A Tivoli Decision Support for z/OS task that includes maintaining the database, updating environment information, and ensuring the accuracy of data collected.

administration dialog

A set of host windows used to administer Tivoli Decision Support for z/OS.

C

collect A process used by Tivoli Decision Support for z/OS to read data from input log data sets, interpret records in the data set, and store the data in DB2 tables in the Tivoli Decision Support for z/OS database.

component

An optionally installable part of a Tivoli Decision Support for z/OS feature. Specifically in Tivoli Decision Support for z/OS, a component refers to a logical group of objects used to collect log data from a specific source, to update the Tivoli Decision Support for z/OS database using that data, and to create reports from data in the database.

control table

A predefined Tivoli Decision Support for z/OS table that controls results returned by some log collector functions.

D

data table

A Tivoli Decision Support for z/OS table that contains performance data used to create reports.

uninstall

An administration dialog option to remove a component from the list of installed components. This action involves deleting from Tivoli Decision Support for z/OS system tables all definitions that the component uses.

E

environment information

All of the information that is added to the log data to create reports. This

information can include data such as performance groups, shift periods, installation definitions, and so on.

H

header

An element of a log definition. Lists fields common to all records in the log.

L

log collector

A Tivoli Decision Support for z/OS program that processes log data sets and provides other Tivoli Decision Support for z/OS services.

log collector language

Tivoli Decision Support for z/OS statements used to supply definitions to and invoke services of the log collector.

log data set

Any sequential data set that is used as input to Tivoli Decision Support for z/OS.

log definition

The description of a log data set processed by the log collector.

log procedure

A program module that is used to process all record types in certain log data sets.

lookup expression

An expression that specifies how a value is obtained from a lookup table.

lookup table

A Tivoli Decision Support for z/OS DB2 table that contains grouping, translation, or substitution information.

P

purge condition

Instruction for purging old data from the database.

R

record definition

The description of a record type contained in the log data sets used by Tivoli Decision Support for z/OS, including detailed record layout and data formats.

record procedure

A program module that is called to process some types of log records.

record type

The classification of records in a log data set.

repeated section

A section of a record that occurs more than once, with each occurrence adjacent to the previous one.

report definition language

Tivoli Decision Support for z/OS statements used to define reports and report groups.

report group

A collection of Tivoli Decision Support for z/OS reports that can be referred to by a single name.

reporting dialog

A set of host or workstation windows used to request reports.

resource group

A collection of network resources that are identified as belonging to a particular department or division. Resources are organized into groups to reflect the structure of an organization.

resource information

Environment information that describes the elements in a network.

S**section**

A structure within a record that contains one or more fields and may contain other sections.

source In an update definition, the record or DB2 table that contains the data used to update a Tivoli Decision Support for z/OS DB2 table.

system table

A DB2 table that stores information that controls log collector processing, Tivoli Decision Support for z/OS dialogs, and reporting.

T**table definition**

Stores data in DB2. It identifies the

database and tablespace in which a table resides, and identifies columns in the table.

target In an update definition, the DB2 table in which Tivoli Decision Support for z/OS stores data from the source record or table.

Tivoli Decision Support for z/OS database

A set of DB2 tables that includes data tables, lookup tables, system tables, and control tables.

timestamp

An element of a log definition. Describes how to derive the timestamp of a record from fields in the header.

U**update definitions**

Instructions for entering data into DB2 tables from records of different types or from other DB2 tables.

V

view An alternative representation of data from one or more tables. A view can include all or some of the columns contained in the table on which it is defined.

Bibliography

Tivoli Decision Support for z/OS publications

Administration Guide and Reference, SH19-6816, SH19-6816
Resource Accounting for z/OS, SH19-4495, SH19-4495
AS/400 System Performance Feature Guide and Reference, SH19-4019, SH19-4019
CICS Performance Feature Guide and Reference, SH19-6820, SH19-6820
Distributed Systems Performance Feature Guide and Reference, SH19-4018, SH19-4018
Guide to Reporting, SH19-6842, SH19-6842
IMS Performance Feature Guide and Reference, SH19-6825, SH19-6825
Language Guide and Reference, SH19-6817, SH19-6817
Messages and Problem Determination, SH19-6902, SH19-6902
Network Performance Feature Installation and Administration, SH19-6901, SH19-6901
Network Performance Feature Reference, SH19-6822, SH19-6822
Network Performance Feature Reports, SH19-6821, SH19-6821
Resource Accounting, SH19-6818, SH19-6818
System Performance Feature Guide, SH19-6819, SH19-6819
System Performance Feature Reference Volume I, SH19-4494, SH19-4494
System Performance Feature Reference Volume II, SC23-7966, SC23-7966

DB2 publications

IBM DB2 Universal Database for z/OS: Utility Guide and Reference, SC18-7427
IBM DB2 Universal Database for z/OS: Messages, GC18-9602
IBM DB2 Universal Database for z/OS: SQL Reference, SC18-7426
IBM DB2 Universal Database for z/OS: Administration Guide, SC18-7413

Index

A

abbreviation
 adding to an update definition 187
 deleting from an update definition 188
accessibility xi, 309
adding a column to a table 181
adding a log data set for collection 214
adding a log ID and collect statements data set 211
adding an abbreviation to an update definition 187
adding an object to a component 138
ADMDEFS nickname ddname, description 113
ADMGDF ddname, graphic reports data set 113
administering lookup and control tables 112
administering problem records 121
administering reports 112
administering the Tivoli Decision for z/OS Support for z/OS database 95
administering user access to tables 202
administration dialog
 collecting data 85
 collecting log data 86
 commands 279
 introduction 10
 options
 Administration window 273
 Components window 274
 Logs window 276
 Primary Menu 273
 Tables window 277
ADMINISTRATION parameter 27
Administration window options
 Help pull-down 274
 Other pull-down
 DB2I option 111
 Exit option 274
 ISPF/PDF option 274
ADMUCDSO, graphic reports to page segments 120
AGGR_VALUE control table 245
allocating libraries in the generic logon procedure
 SYSPROC
 DRL.LOCAL.EXEC 27
allocation overview, ddname 69
APPLY SCHEDULE clause
 modifying 189
authorization ID, DB2 secondary 18
AVAILABILITY_D,_W,_M 247
AVAILABILITY_PARM lookup table 250
AVAILABILITY_T 248

B

backing up the Tivoli Decision Support for z/OS database 105
backup, incremental-image or full-image 107
batch
 generating problem records in 121
 running reports 112
batch mode
 collect 86
 installing component 130
 installing component in 130
 reporting 120
batch print SYSOUT class dialog parameter 61
BookMaster documents, including saved charts in 120
books ix, xi

C

calculating and monitoring table space requirements 97
calculating table space requirements monitoring, and 97
Changing or deleting rows using the QMF table editor 166
changing space definitions 39
changing the collect statements data set name 211
changing the retention period of information about a log data set 220
CICS control tables
 CICS_DICTIONARY 246
 CICS_FIELD 246
CICS Partitioning feature
 customizing 25
collect
 batch mode 86
 COLLECT log collector language statement 86
 deciding which data sets to 215
 generic collect job 86
 improving performance 94
 log collector messages 90
 monitoring activity 90
 network configuration data collect job 89
 vital product data collect job 89
collect activity
 monitoring 90
collect messages
 sample 90
 using 90
collect performance
 improving 94
collect process 7
collect statements
 changing data set name 211
 editing 209

collect statements (*continued*)
 Tivoli Decision Support for z/OS supplied, modifying 210
collect statements data set
 adding 211
collect statements for log data manager
 listing data sets containing 209
Collect Statistics window 92
Collect window 36
collected data sets
 viewing information about 220
collecting data
 through administration dialog 85
collecting data from a log into DB2 tables 146
collecting data from IMS 89
collecting data from Tivoli Information Management for z/OS 89
collecting log data 85
collecting multiple log data sets 93
collecting network configuration data 89
column
 adding to a table 181
 column definition
 displaying and modifying 180
commands and options, administration dialog 273
common data tables 246, 247
 AVAILABILITY_D,_W,_M 247
 AVAILABILITY_T 248
 EXCEPTION_T 249
 MIGRATION_LOG 250
 retention periods 247
 summarization level 247
COMPONENT command 279
component
 adding an object 138
 creating 140
 deleting 140
 deleting object 138
 installation
 definition members 71
 installing and uninstalling 126
 installing online 128
 Sample component
 definition member 71
 description 253
 uninstalling 134
 viewing objects 137
component definition
 working with 136
component installation
 excluding and object from 139
 including an object 139
components
 installing 40
Components window 126
Components window options 274
 Component pull-down
 Print list option 275
 Help pull-down 276

- Components window options *(continued)*
 - Other pull-down
 - DB2I option 111
 - ISPF/PDF option 274
 - Space pull-down 39
- concatenation
 - of log data sets 215
- concatenation of log data sets 215
- considerations when running
 - DRLJTBSR 100
- control and common tables 243
 - AGGR_VALUE 245
 - AVAILABILITY_PARM 250
 - CICS control tables
 - CICS_DICTIONARY 246
 - CICS_FIELD 246
 - common data tables 247
 - DAY_OF_WEEK 243
 - description, lookup tables 250
 - PERIOD_PLAN 244
 - SCHEDULE 244
 - SPECIAL_DAY 245
 - USER_GROUP 251
- control and lookup tables
 - administering 112
- controlling objects that you have
 - modified 136
- conventions
 - typeface xii
- correcting corrupted data in the Tivoli Decision Support for z/OS database 108
- correcting out-of-space condition in table or index space 107
- corrupted data in the Tivoli Decision Support for z/OS database
 - correcting 108
- create invoices and reports
 - running DRLNJOB3 47
- create system tables 33
- creating a component 140
- creating a log definition 153
- creating a record definition 159
- creating a record procedure
 - definition 161
- creating a report on a record 150
- creating a table 198
- creating a table space 200
- Creating a table using an existing table as a template 199
- creating an update definitions 201
- creating and updating system tables with a batch job 35
- creating report groups 120
- customer support 312
- customizing
 - DRLEINI1 27
 - generic logon procedure 27
 - JCL sample jobs 35
- customizing Usage and Accounting Collector
 - execute DRLNINIT 42

D

- data
 - collecting from IMS 89

- data *(continued)*
 - collecting from Tivoli Information Management for z/OS 89
- data backup, incremental-image or full-image 107
- data collecting
 - batch collect 89
 - IMS 89
- data from Tivoli Information Management for z/OS
 - collecting 89
- data in tables
 - working with 164
- data security 18
 - controlling 95
 - initializing 21
- Data Selection window 36
- data set
 - prefix dialog parameter 61
 - saving a table definition in 197
- data sets 17
 - deciding which to collect 215
- data sets collected
 - viewing list 144
- data tables, common 246, 247
 - AVAILABILITY_D, _W, _M 247
 - AVAILABILITY_T 248
 - EXCEPTION_T 249
 - MIGRATION_LOG 250
 - retention periods 247
 - summarization level 247
- database
 - access 110
 - administration 95
 - backing up 105, 107
 - error recovery 107
 - initialization 21
 - introduction 9
 - monitoring size 108
 - name dialog parameter 59
 - security 18, 110
 - tools 111
- database access
 - monitoring 110
- database backup
 - determining when 107
- database security
 - maintaining 110
- date set
 - viewing dump 220
- DAY_OF_WEEK control table 243
- DB2
 - data sets prefix dialog parameter 61
 - DB2 plan name for TDS 59
 - how Tivoli Decision Support for z/OS uses 96
 - locking and concurrency 109
 - messages
 - during system table creation 33
 - performance 21
 - statistics 108
 - subsystem name dialog parameter 59
 - tools 111
- DB2 concepts
 - understanding 95
- DB2 High Performance Unload
 - integration 176

- DB2 High Performance Unload utility
 - running 177
 - sample control statement 178
- DB2 tables
 - collecting data from a log into 146
- DB2 utility
 - RUNSTATS 167
- DB2I
 - concepts 96
 - DB2I Primary Option Menu 111
 - secondary authorization IDs 21
 - statistics 108
 - Tivoli Decision Support for z/OS interaction 96
 - tools 111
- DB2I command 279
- DCOLLECT records 267
- DEBUG parameter 27
- deciding which log data sets to collect 215
- DEFINE LOG log collector language statement 5, 74
- DEFINE RECORD
 - log collector language statement 6
- DEFINE RECORD log collector language statement 74
- DEFINE UPDATE 187
 - log collector language statement 6
- DEFINE UPDATE log collector language statement 75
- defining objects, overview 71
- defining reports 78
- defining table spaces and indexes using the GENERATE statement 77
- defining updates and views 77
- definition members 71, 75
 - component definitions 71
 - DRLxxx.SDRLEDFS library 73
 - feature 73, 74
 - installation order 73
 - log 74
 - record 74
 - report 78
 - Sample component definition member 73
 - table and update definition members 75
 - table space 75
- deleting a component 140
- deleting a log data set 146
- deleting a log definition 154
- deleting a record definition 160
- deleting a record procedure
 - definition 162
- deleting a table index 184
- deleting a table or view 200
- deleting an abbreviation from an update definition 188
- deleting an object from a component 138
- deleting an update definition 202
- deleting information about a log data set 214
- deleting the information about a log data set 220, 222
- detail tables
 - AVAILABILITY_T 248

detail tables (*continued*)
 EXCEPTION_T 249
 MIGRATION_LOG 250
 determining partitioning mode and keys 40
 determining when to back up the Tivoli Decision Support for z/OS database 107
 dialog
 commands 279
 Dialog Parameters window 29, 56
 when QMF is not used 57
 DRLEINI1 initialization exec 27
 language options 61
 parameters 29, 33
 preparing 27
 dialog parameters 55
 variables and fields 58
 Dialog Parameters window 29
 Dialog Parameters window 57
 overview 56
 QMF not used 57
 when QMF is used 56
 disability 309
 DISPLAY RECORD record_type command 279
 DISPLAY REPort report_ID command 279
 DISPLAY report_ID command 279
 DISPLAY TABLE table_name command 279
 DISPLAY table_name command 279
 displaying a view definition 196
 displaying and adding a table index 182
 displaying and editing the purge condition of a table 190
 displaying and modifying a column definition 180
 displaying and modifying a table or index space 192
 displaying and modifying update definitions of a table 184
 displaying log statistics 148
 displaying the contents of a log 149
 displaying the contents of a table 164
 displaying update definitions associated with a record 159
 distribution clause
 modifying 188
 documentation
 TDS 321
 DRL.LOCAL.CHARTS 113
 DRL.LOCAL.DEFS definitions library 63
 DRL.LOCAL.EXEC, allocating 27
 DRL.LOCAL.REPORTS 113
 DRL.LOCAL.USER.DEFS definitions library 63
 DRL182.SDRLDEFS
 naming convention for members 81
 DRL182.SDRLRENU
 naming convention for members 81
 DRLCHARTS system table 235
 DRLCOLUMNS view on DB2 catalog 241
 DRLCOMP_OBJECTS system table 236
 DRLCOMP_PARTS system table 236
 DRLCOMPONENTS system table 236
 DRLEINI1 listing 55
 DRLELDMC
 sample job 216
 DRLESTRA command 279
 DRLEXPRESSIONS
 system table 225
 DRLFIELDS
 system table 225
 DRLGROUP_REPORTS system table 237
 DRLGROUPS system table 237
 DRLINDEXES view on DB2 catalog 241
 DRLINDEXPART view on DB2 catalog 241
 DRLJBATR batch reporting job 113
 DRLJCOIM
 IMS collect job 89
 DRLJCOIN collect job 89
 DRLJCOLL generic collect job 86
 DRLJCOVP network VPD collect job 89
 DRLJDBIN job 22
 DRLJDBIP
 database initialization job 26
 DRLJDBIP job
 CICS Partitioning feature 25
 DRLJEXCE problem record job 121
 DRLJLDMC
 setting the parameters for job 218
 DRLJLDMC collect job
 parameters it uses 215
 DRLJLDMML
 job step, using 206
 sample job 207
 setting the parameters for 208
 DRLJLDMML sample job 207
 DRLJPURG purge job 104
 DRLJRFT report format table 89
 DRLJRUNS RUNSTATS job 108
 DRLKEYS view on DB2 catalog 241
 DRLLDL_COLLECTSTMT
 system table 226
 DRLLDL_LOGDATASETS
 system table 227
 DRLLOGDATASETS
 system table 227
 DRLLOGDATASETS system table 92
 DRLLOGS
 system table 228
 DRLNINIT
 executing 42
 DRLNJOB2
 processing SMF data, using 45
 DRLOBJECT_DATA view on Q.OBJECT_DATA 241
 DRLPURGECOND
 system table 229
 DRLRECORDPROCS
 system table 229
 DRLRECORDS
 system table 230
 DRLREP dname, tabular reports data set 113
 DRLREPORT_ATTR system table 238
 DRLREPORT_COLUMNS system table 239
 DRLREPORT_QUERIES system table 239
 DRLREPORT_TEXT system table 239
 DRLREPORT_VARS system table 240
 DRLREPORTS system table 237
 DRLRPROCINPOT system table 230
 DRLSEARCH_ATTR system table 240
 DRLSEARCHES system table 240
 DRLSECTIONS
 system table 230
 DRLTABAUTH view on DB2 catalog 241
 DRLTABLEPART view on DB2 catalog 241
 DRLTABLES view on DB2 catalog 241
 DRLTABLESPACE view on DB2 catalog 241
 DRLUPDATECOLS
 system table 231
 DRLUPDATEDISTR
 system table 231
 DRLUPDATELETS
 system table 231
 DRLUPDATES system table 232
 DRLUSER_GROUPPREPS 241
 DRLUSER_GROUPS 241
 DRLUSER_REPORTATTR 241
 DRLUSER_REPORTCOLS 242
 DRLUSER_REPORTQRYs 242
 DRLUSER_REPORTS 242
 DRLUSER_REPORTTEXT 242
 DRLUSER_REPORTVARS 242
 DRLUSER_SEARCHATTR 242
 DRLUSER_SEARCHES 242
 DRLVIEWS view on DB2 catalog 241
 DRLxxx.SDRLDEFS definitions library
 definition members 73
 DRLxxx.SDRLEXEC, allocating 27
 DRLxxx.SDRLOAD, allocating in the logon procedure 27
 DSNxxx.SDSNLOAD, allocating in the logon procedure 27
 dump data set
 viewing 220, 221
 DYNAMNBR
 setting the value 218

E

editing
 object definition 137
 editing a table using the ISPF editor 167
 editing the collect statements 209
 editing the contents of a table 165
 education xi
 EREP
 records shipped with Tivoli Decision Support for z/OS 268
 errors, recovering from database 107
 EXCEPTION_T detail table 249
 exceptions
 reviewing 121
 exceptions and problem records 121
 excluding an object from a component installation 139
 exporting table data to an IXF file 172

F

- features, Tivoli Decision Support for
 - z/OS performance
 - definition member description 73, 74
 - record 74
 - update and view 77
 - installation with base 17
- fixes, obtaining 311
- flow of Tivoli Decision Support for z/OS data 8

G

- GDDM
 - allocating load library 28
 - GDDM-PGF
 - ADMUCDSO, graphic reports to
 - page segments 120
 - formats data set 63
 - local formats data set 63
 - nicknames, ADMDEFS ddname 113
 - GDDM.SADMMOD, allocating in the logon procedure 27
- GENERATE statement
 - defining table spaces and indexes 77
- GENERATE_KEYS 234
- GENERATE_PROFILES 233
- generating problem records in batch 121
- generic logon procedure, customizing 27
- graphic reports
 - BookMaster documents, in 120
 - data set ddname, ADMGDF 113
 - allocation overview 69
 - dialog parameter description 64
 - page segments from, making 120

H

- hardware
 - prerequisites 15
- header fields
 - working with 152
- HELP command 279

I

- importing the contents of an IXF file to a table 171
- improving collect performance 94
- including an object in a component installation 139
- index space
 - displaying and modifying 192
 - making changes 194
 - out of space 107
- index space definitions 39
- index space, out of space condition 107
- Indexes window options
 - Utilities pull-down
 - Run DB2 REORG utility 96
- INFO command 279
- INFO SEArch command 279
- INFO SEArch search-argument command 280

- information centers, searching for problem resolution 311
- installation
 - base product and feature
 - installation 17, 36
 - description of DRLEINI1 variables 64
 - DRLEINI1 listing, variables 55
 - hardware prerequisites 15
 - software prerequisites 16
 - installation prerequisites 15
 - installation reference 55
 - installing
 - feature (separate SMP/E job) 50
 - Usage and Accounting Collector 41
 - installing a component 126
 - installing and uninstalling a component 126
 - installing components 40
 - installing other Tivoli Decision Support for z/OS systems 49
 - installing the component in batch mode 130
 - installing the component online 128
 - installing Tivoli Decision for z/OS
 - features
 - separately 50
 - installing Tivoli Decision Support for z/OS 15
 - determining partitioning mode and keys 40
 - installing components 40
 - reviewing results of SMP/E installation 17
 - installing Usage and Accounting Collector 41
 - DRLNJOB1 44
 - DRLNJOB3 (DRLCMONY), running 47
 - JCL
 - customizing Usage and Accounting Collector 42
 - processing Usage and Accounting Collector Subsystems 49
 - integration with DB2 High Performance Unload 176
 - Internet
 - searching for problem resolution 311
 - introduction to the Key Performance Metrics Components 12
 - invoking the log data manager 205
 - ISPF
 - ISPF.PROFILE 27
 - ISPF command 280
 - ISPF editor
 - editing a table 167
 - IXF
 - exporting table data to 172
 - IXF file
 - importing contents to a table 171
- J**
 - JCL sample jobs
 - customizing 35
 - job statement information dialog parameter 64

- job step for recording a log data set for collection 206

K

- Key Performance Metrics Components
 - introduction 12
- knowledge bases, searching for problem resolution 311

L

- language-dependent data sets 17
- library allocation in the generic logon procedure
 - STEPLIB
 - DRLxxx.SDRLOAD 27
 - SYSEXEC 27
- library definition members, DRLxxx.SDRLDEFS 73
- listing
 - subset of tables in the Tables window 198
- listing a subset of tables in the Tables window 198
- listing the log data sets to be collected 212
- loading tables 172
- local data sets 17
- local definitions data set dialog parameter 63
- local messages data set dialog parameter 63
- local user definitions data set dialog parameter 63
- LOcate argument command 280
- locking and concurrency 109
- log and record definitions
 - working with 143
- log and record procedures 6
- log collector
 - introduction 5
 - modifying statements 209
 - system tables 225
- log collector language
 - COLLECT 86
 - DEFINE LOG 5, 74
 - DEFINE RECORD 74
 - DEFINE UPDATE 75
 - SQL 71
 - SQL CREATE 75
- log data
 - collecting 85
- log data manager
 - invoking 205
 - listing log data sets containing collect statements 209
 - modifying list of log data sets to be collected 212
 - summary of use of 205
- log data manager option
 - working with 205
- log data set
 - adding for collection 214
 - changing the retention period of information 220

- log data set (*continued*)
 - deleting 146
 - deleting information about 214, 220, 222
 - modifying log ID 213
 - recording for re-collection 214
 - recording to be collected again 222
 - viewing unsuccessfully collected 221
- log data sets
 - concatenation of 215
 - listing sets to be collected 212
 - modifying list of successfully collected 219
 - modifying the list of unsuccessfully collected 221
 - viewing information about successfully collected 220
 - viewing list collected 144
- log definition
 - creating 153
 - deleting 154
 - viewing and modifying 152
- log definitions
 - defining a log 74
 - introduction 5
 - working with 151
- log ID
 - adding collect statements data set 211
- log statistics 92
 - displaying 148
- logon procedure, customizing 27
- logs
 - working with the contents of 144
- LOGS command 280
- Logs window options 276
 - Help pull-down 277
 - Log pull-down
 - Exit option 276
 - Print list option 276
 - Save definition option 276
 - Other pull-down
 - DB2I option 111
 - ISPF/PDF option 275
 - View pull-down
 - All option 276
 - Some option 276
- LOGSTAT, log data set statistics 92, 148
- lookup and control tables 243
 - administering 112
 - AGGR_VALUE 245
 - AVAILABILITY_PARM 250
 - CICS control tables
 - CICS_DICTIONARY 246
 - CICS_FIELD 246
 - common data tables 247
 - DAY_OF_WEEK 243
 - description, lookup tables 250
 - PERIOD_PLAN 244
 - SCHEDULE 244
 - SPECIAL_DAY 245
 - USER_GROUP 251

M

- maintaining database security 110
- making changes to an index space 194

- making table space parameter changes that do not require offline or batch action 195
- manuals ix, xi
 - TDS 321
- marking objects user-modified 136
- messages
 - collect 90
 - DB2
 - system table creation 33
- migration of objects, using VERSION variable 72
- MIGRATION_LOG detail table 250
- modifiable area of DRLEINI1 55
- modifying a distribution clause 188
- modifying an apply schedule clause 189
- modifying log collector statements 209
- modifying the list of successfully collected log data sets 219
- modifying the list of unsuccessfully collected log data sets 221
- modifying the log ID for a log data set 213
- modifying Tivoli Decision Support for z/OS supplied collect statements 210
- monitoring collect activity 90
- monitoring database access 110
- monitoring size of the Tivoli Decision Support for z/OS database 108
- monitoring table space requirements calculating, and 97
- multiple log data sets
 - collecting 93
- multiple Tivoli Decision Support for z/OS, installing 49

N

- naming convention for members of DRL182.SDRLEDFS 81
- naming convention for members of DRL182.SDRLEDFU 81
- naming convention for Tivoli Decision Support for z/OS definition members 81
- navigation-administration dialog options and commands 273
- network
 - collecting configuration data 89
 - network configuration data collecting 89
 - network data collect job 89
- nonsummarized data tables 248, 249, 250

O

- object
 - excluding from a component installation 139
 - including in a component installation 139
- object definition
 - viewing or editing 137
- object definitions 71

- objects
 - controlling 136
 - overview 71
 - viewing in a component 137
- online publications
 - accessing xi
- OPC/ESA
 - records shipped 269
- opening a table to display columns 179
- operating routines
 - setting up 85
- options and commands, administration dialog 273
- ordering publications xi
- out of space condition
 - correcting 107
- output options for reports 112
- overview of defining objects 71, 86
- overview of Dialog Parameters window 56
- overview of Tivoli Decision Support for z/OS data flow 8

P

- parameters
 - setting for job DRLJLDMC 218
 - table space reporting 99
- parameters for table space reporting 99
- partitioning mode and keys
 - determining 40
- PDF command 280
- performing routine data collection 89
- PERIOD_PLAN control table 244
- prefix for all other tables dialog parameter 59
- prefix for system tables dialog parameter 60
- Primary Menu options
 - Help pull-down 273
 - Options pull-down
 - Dialog parameters option 58
- printed reports from batch 113
- Printer line count per page dialog parameter 61
- printing a list of tables Tivoli Decision Support for z/OS tables 197
- problem determination
 - describing problems 314
 - determining business impact 313
 - submitting problems 314
- problem records
 - administering 121
 - generating 121
- procedures
 - log and record 6
- processing SMF data
 - using DRLNJOB2 (DRLCDATA and DRLCACCT) 45
- processing Usage and Accounting Collector subsystems 49
- publications ix
 - accessing online xi
 - DB2 321
 - ordering xi
 - TDS 321

- pull-down options
 - Administration window 273
 - Components window 274
 - Logs window 276
 - Primary Menu 273
 - Tables window 277

- purge condition
 - displaying and editing 190
- Purge utility 104
- purging a table 172
- purging data 104

Q

- Q.OBJECT_DATA QMF control table, view of 241

QMF

- batch reporting 120
- data sets prefix dialog parameter 62
- initialization 32
- language option dialog parameter 61
- query 32
- query, importing 32
- setup 32
- view on objects table 241

- QMF command 280

QMF table editor

- changing or deleting rows 166

- QMFxxx.SDSQLOAD, allocating in the logon procedure 27

query

- modifying to eliminate report variables 113
- typical report 113

R

RACF

- records shipped 269

- recalculating the content of a table 169

record

- creating a report 150

record definition

- creating 159
- deleting 160
- viewing and modifying 155
- working with sections 158

record definition fields

- working with 157

record definitions

- DEFINE RECORD log collector

- language statement 74

- definition members 74, 78

- introduction 6, 74

- record definitions shipped with Tivoli

- Decision Support for z/OS 259

- DCOLLECT 267

- EREP 268

- IMS SLDS 264

- OPC 269

- SMF 259

- Tivoli Workload Scheduler for

- z/OS (OPC) 269

- VM accounting 269

- VMPRF 270

- record definitions in a log

- working with 154

- Record Definitions window options

- Other pull-down

- DB2I option 111

- ISPF/PDF option 275

- record procedure definition

- creating 161

- viewing and modifying 160

- record procedure definitions

- deleting 162

- recording a log data set

- job step for 206

- recording a log data set for

- re-collection 214

- recording a log data set to be collected again 222

records

- Linux on zSeries 268

- recovering from database errors 107

- reference, installation 55

- Reorg/Discard utility 101

report

- creating on a record 150

- report definition language, defining

- report groups 78

- report format table, DRLJRFT 89

- reporting dialog

- introduction 11

- reporting dialog mode dialog

- parameter 61

reports

- PRA001 281, 286

- PRA002 282

- PRA003 284

- PRA004 285

- PRA006 287

- reports and report groups 79

- adding to report group 120

- administering 112

- administration 113, 119

- batch creation 117

- creating groups 120

- customizing for batch processing 113

- defining 79

- examples 78

- graphic reports 113

- output options 112

- print options 112

- printing or saving in batch 113

- QMF batch reporting 120

- query example 113

- Reports window 36

- running in batch 112, 117

- saved reports 113

- saving in batch 113, 119, 120

- REPORTs command 280

- REPORTS parameter 27

- RESET parameter 27

- retention periods, common data

- tables 247

- reviewing DB2 parameters 39

- reviewing exceptions and generating

- problem records 121

- reviewing Key Performance Metrics table

- space profiles prior to installation 135

- reviewing the GENERATE statements for table spaces, tables, and indexes 135

- reviewing the results of the SMP/E

- installation 17

- routine data collection

- performing< 89

- routines

- performing data collection 89

- running DB2 High Performance Unload

- utility 177

- running DRLJTBSR

- considerations 100

- running DRLNJOB3 to create invoices

- and reports 47

- running reports in batch 112

- RUNSTATS utility 167

S

- sample collect messages 90

Sample component

- component definition member 71

- description 253

- object definition members 71

- Sample Report 1 255

- Sample Report 2 256

- Sample Report 3 257

- SAMPLE_H,_M tables 254

- SAMPLE_USER lookup table 254

Sample components reports

- introduction 254

- sample JCL jobs 35

sample job

- DRLELDMC 216

- DRLJLDML 207

SAMPLE log type

- collecting log data 85

- saved charts data set dialog

- parameter 64

- saved reports data set dialog

- parameter 63

- saved reports, batch creation 119

- saving a table definition in a data

- set 197

- SCHEDULE control table 244

- secondary authorization IDs

- security without 20

- sections in a record definition

- working with 158

security

- without secondary authorization

- IDs 20

- security without secondary authorization

- IDs 20

- security, database 18

- secondary authorization IDs 18

- setting the DYNAMNBR value 218

- setting the parameters for job

- DRLJLDMC 218

- setting the parameters for job

- DRLJLDML 208

- show TDS environment data 60

- showing the size of the table 167

- sing the administration dialog to delete a

- column from a table 200

- SLDS records 264

- SMF records 259

- SMF_VPD data collect 89
- SMP/E installation 17, 50
 - reviewing results 17
- software
 - prerequisites 16
- Software Support
 - contacting 312
 - describing problems 314
 - determining business impact 313
 - receiving weekly updates 312
 - submitting problems 314
- SOrt column_name | position ASC | DES
 - command 280
- SPECIAL_DAY control table 245
- SQL ID to use (in QMF) dialog
 - parameter 60
- SQL log collector language statement 71
- SQLMAX dialog parameter 61
- storage group default dialog
 - parameter 59
- subset of tables
 - listing in the Tables window 198
- successfully collected data sets
 - modifying list of 219
- SYSOUt class (in QMF) dialog
 - parameter 61
- SYSPROC
 - DRL.LOCAL.EXEC 27
- SYStem command 280
- system table
 - DRLEXPRESSIONS 225
 - DRLFIELDS 225
- system tables
 - log collector 225
- system tables and views
 - creating system tables 33
 - dialog system tables 235
 - DRLCHARTS 235
 - DRLCOLUMNS 241
 - DRLCOMP_OBJECTS 236
 - DRLCOMP_PARTS 236
 - DRLCOMPONENTS 236
 - DRLGROUP_REPORTS 237
 - DRLGROUPS 237
 - DRLINDEXES 241
 - DRLINDEXPART 241
 - DRLKEYS 241
 - DRLLDM_COLLECTSTMT 226
 - DRLLDM_LOGDATASETS 227
 - DRLLOGDATASETS 92, 227
 - DRLLOGS 228
 - DRLOBJECT_DATA 241
 - DRLPURGECOND 229
 - DRLRECORDPROCS 229
 - DRLRECORDS 230
 - DRLREPORT_ATTR 238
 - DRLREPORT_COLUMNS 239
 - DRLREPORT_QUERIES 239
 - DRLREPORT_TEXT 239
 - DRLREPORT_VARS 240
 - DRLREPORTS 237
 - DRLRPROCINPUT 230
 - DRLSEARCH_ATTR 240
 - DRLSEARCHES 240
 - DRLSECTIONS 230
 - DRLTABAUTH 241
 - DRLTABLEPART 241

- system tables and views (*continued*)
 - DRLTABLES 241
 - DRLTABLESPACE 241
 - DRLUPDATECOLS 231
 - DRLUPDATEDISTR 231
 - DRLUPDATELETS 231
 - DRLUPDATES 232
 - DRLUSER_GROUPPREPS 241
 - DRLUSER_GROUPS 241
 - DRLUSER_REPORTATTR 241
 - DRLUSER_REPORTCOLS 242
 - DRLUSER_REPORTQRYs 242
 - DRLUSER_REPORTS 242
 - DRLUSER_REPORTTEXT 242
 - DRLUSER_REPORTVARS 242
 - DRLUSER_SEARCHATTR 242
 - DRLUSER_SEARCHES 242
 - DRLVIEWS 241
 - GENERATE_KEYS 234
 - GENERATE_PROFILES 233
 - updating system tables 50
- System Tables window 33, 50
- system window
 - Tivoli Decision Support for z/OS
 - administration dialog windows
 - system window 29
- systems, installing other Tivoli Decision Support for z/OS 49

T

- table
 - adding a column 181
 - creating 198
 - deleting 200
 - deleting index 184
 - displaying and adding index 182
 - displaying and editing purge
 - condition of 190
 - displaying and modifying 184
 - displaying contents of 164
 - editing contents of 165
 - opening to display columns 179
 - purging 172
 - recalculating contents of 169
- table access 203
- table and update definitions
 - creating
 - system tables 33
 - definition members 75
 - introduction 75
 - IXF files, importing 171
 - lookup and control tables 112
 - modifying an APPLY SCHEDULE
 - clause 189
- TABLE command 280
- table data
 - exporting to an IXF file 172
- table definition
 - saving in a data set 197
- table definitions
 - introduction 6
- table space 75
 - backing up 107
 - creating 200
 - definition members 75
 - displaying and modifying 192

- table space (*continued*)
 - introduction 10
 - making parameter changes that do not
 - require offline or batch action 195
 - out of space 107
- table space definitions 39
- table space profiles
 - reviewing Key Performance Metrics
 - prior to installation 135
 - working with 134
- table space reporting
 - parameters 99
- table space, out of space condition 107
- table spaces
 - understanding 96
- table spaces, tables and indexes
 - reviewing the GENERATE
 - statements 135
- table summarization levels,
 - common 247
- tables
 - administering user access to 202
 - unloading and loading 172
- tables and update definitions
 - working with 163
- tables and views
 - GENERATE_PROFILES 233
- tables and views, system 225
 - creating system tables 33
 - dialog system tables 235
 - DRLCHARTS 235
 - DRLCOLUMNS 241
 - DRLCOMP_OBJECTS 236
 - DRLCOMP_PARTS 236
 - DRLCOMPONENTS 236
 - DRLGROUP_REPORTS 237
 - DRLGROUPS 237
 - DRLINDEXES 241
 - DRLINDEXPART 241
 - DRLKEYS 241
 - DRLLOGDATASETS 92
 - DRLOBJECT_DATA 241
 - DRLREPORT_ATTR 238
 - DRLREPORT_COLUMNS 239
 - DRLREPORT_QUERIES 239
 - DRLREPORT_TEXT 239
 - DRLREPORT_VARS 240
 - DRLREPORTS 237
 - DRLSEARCH_ATTR 240
 - DRLSEARCHES 240
 - DRLSECTIONS 230
 - DRLTABAUTH 241
 - DRLTABLEPART 241
 - DRLTABLES 241
 - DRLTABLESPACE 241
 - DRLUPDATECOLS 231
 - DRLUPDATEDISTR 231
 - DRLUPDATELETS 231
 - DRLUPDATES 232
 - DRLUSER_GROUPPREPS 241
 - DRLUSER_GROUPS 241
 - DRLUSER_REPORTATTR 241
 - DRLUSER_REPORTCOLS 242
 - DRLUSER_REPORTQRYs 242
 - DRLUSER_REPORTS 242
 - DRLUSER_REPORTTEXT 242
 - DRLUSER_REPORTVARS 242

- tables and views, system *(continued)*
 - DRLUSER_SEARCHATTR 242
 - DRLUSER_SEARCHES 242
 - DRLVIEWS 241
 - GENERATE_KEYS 234
 - updating system tables 50
- tables naming standard, common 247
- Tables Window
 - listing a subset of tables 198
- Tables window options 277
 - Help pull-down 279
 - Other pull-down
 - ISPF/PDF option 276
 - Table pull-down
 - Exit option 277
- tables, control and common 243
 - AGGR_VALUE 245
 - AVAILABILITY_PARM 250
 - CICS control tables
 - CICS_DICTIONARY 246
 - CICS_FIELD 246
 - common data tables 247
 - DAY_OF_WEEK 243
 - description, lookup tables 250
 - PERIOD_PLAN 244
 - SCHEDULE 244
 - SPECIAL_DAY 245
 - USER_GROUP 251
- tables, control and lookup
 - administering 112
- tabular reports, DRLREP ddname 113
- temporary data sets prefix dialog
 - parameter 62
- testing component
 - verify proper installation 133
- testing the component to verify its proper
 - installation 133
- the DRLJLDMC collect job and the
 - parameters it uses 215
- timestamp tables
 - AVAILABILITY_T 248
 - EXCEPTION_T 249
- Tivoli Decision for z/OS
 - objects overview 71
- Tivoli Decision Support for z/OS
 - administration dialog windows 36, 50, 56
 - System Tables window 33
 - Administration window options 273
 - component installation 71
 - data flow 8
 - data sets 17
 - data sets prefix dialog parameter 62
 - database administration 95
 - database, introduction to 9
 - feature installation 50
 - installation 15
 - data sets 17
 - database security 18
 - DB2 database initialization 21
 - personal dialog parameters 29
 - QMF setup 32
 - test 36
 - introducing 3
 - introduction 3
 - performance features 4
 - Primary Menu options 273

- Tivoli Decision Support for z/OS
 - (continued)*
 - record definitions shipped with Tivoli Decision Support for z/OS 259
- Tivoli Decision Support for z/OS
 - administration dialog windows
 - Administration window 29
 - Data Selection window 36
 - Dialog Parameters window 56, 57
 - Logs window 36
 - Primary Menu 29
 - Reports window 36
 - system window 29
- Tivoli Decision Support for z/OS
 - definition members
 - naming convention 81
- Tivoli Decision Support for z/OS dialog
 - options 273
- Tivoli Decision Support for z/OS
 - performance features
 - introduction 4
- Tivoli Decision Support for z/OS Primary
 - Menu 29
- Tivoli Decision Support for z/OS tables
 - printing list 197
- Tivoli Decision Support for z/OS Version
 - variable format 73
- Tivoli software information center xi
- Tivoli technical training xi
- Tivoli Workload Scheduler for z/OS
 - (OPC)
 - records shipped 269
- training, Tivoli technical xi
- typeface conventions xii

U

- understanding DB2 concepts 95
- understanding how Tivoli Decision Support for z/OS uses DB2 96
- understanding how Tivoli Decision Support for z/OS uses DB2 locking and concurrency 109
- understanding table spaces 96
- uninstalling
 - component 134
- uninstalling a component 134
- unloading and loading tables 172
- unloading tables 172
- update definition
 - adding an abbreviation 187
 - creating 201
 - deleting 202
 - deleting an abbreviation 188
- Update Definition window 187
- update definitions 75
 - APPLY SCHEDULE clause 189
 - definition member 77
 - displaying 159
 - displaying and modifying 184
 - introduction 6, 75
- Usage and Accounting Collector
 - installing 41
 - introduction 4
- USER_GROUP lookup table 251
- using collect messages 90
- using the DRLJLDMC job step 206

- utility
 - Purge 104
 - Reorg/Discard 101

V

- variables and fields
 - dialog parameters 58
- variables, eliminating report 113
- verify installation
 - testing the component 133
- VERSION
 - Tivoli Decision Support for z/OS variable format 73
- VERSION variable 72
- view
 - deleting 200
- view definition
 - changing a comment 196
 - displaying 196
- view definitions
 - definition member 77
- viewing
 - object definition 137
- viewing a list of log data sets
 - collected 144
- viewing and modifying a log
 - definition 152
- viewing and modifying a record
 - definition 155
- viewing and modifying a record
 - procedure definition 160
- viewing objects in a component 137
- viewing or editing an object
 - definitions 137
- viewing the dump data set 220, 221
- viewing the information about
 - successfully collected log data sets 220
- viewing the unsuccessfully collected log
 - data set 221
- views and tables, system 225
 - creating system tables 33
 - dialog system tables 235
 - DRLCHARTS 235
 - DRLCOLUMNS 241
 - DRLCOMP_OBJECTS 236
 - DRLCOMP_PARTS 236
 - DRLCOMPONENTS 236
 - DRLGROUP_REPORTS 237
 - DRLGROUPS 237
 - DRLINDEXES 241
 - DRLINDEXPART 241
 - DRLKEYS 241
 - DRLLOGDATASETS 92
 - DRLOBJECT_DATA 241
 - DRLREPORT_ATTR 238
 - DRLREPORT_COLUMNS 239
 - DRLREPORT_QUERIES 239
 - DRLREPORT_TEXT 239
 - DRLREPORT_VARS 240
 - DRLREPORTS 237
 - DRLSEARCH_ATTR 240
 - DRLSEARCHES 240
 - DRLSECTIONS 230
 - DRLTABAUTH 241
 - DRLTABLEPART 241
 - DRLTABLES 241

- views and tables, system *(continued)*
 - DRLTABLESPACE 241
 - DRLUPDATECOLS 231
 - DRLUPDATEDISTR 231
 - DRLUPDATELETS 231
 - DRLUPDATES 232
 - DRLUSER_GROUPPREPS 241
 - DRLUSER_GROUPS 241
 - DRLUSER_REPORTATTR 241
 - DRLUSER_REPORTCOLS 242
 - DRLUSER_REPORTQRYs 242
 - DRLUSER_REPORTS 242
 - DRLUSER_REPORTTEXT 242
 - DRLUSER_REPORTVARS 242
 - DRLUSER_SEARCHATTR 242
 - DRLUSER_SEARCHES 242
 - DRLVIEWS 241
 - GENERATE_PROFILES 233
 - updating system tables 50
- views on DB2 catalog tables 241
- views on Tivoli Decision Support for z/OS system tables 241
- VM accounting records 269
- VMPRF
 - record definitions 270
- VPD data collecting 89

Z

- z/VM Performance Toolkit
 - record definitions 270

W

- windows, administration dialog
 - windows 56, 57
 - Administration window 10, 29
 - Collect Statistics window 92
 - Collect window 36
 - Data Selection window 36
 - Logs window 36
 - Primary Menu 29
 - Reports window 36
 - System Tables window 33, 50
 - system window 29
 - System window 32
- working with a component
 - definition 136
- working with abbreviations 187
- working with components 125
- working with data in tables 164
- working with fields in a record
 - definition 157
- working with header fields 152
- working with log and record
 - definitions 143
- working with log definitions 151
- working with record definitions in a
 - log 154
- working with sections in a record
 - definition 158
- working with table space profiles 134
- working with tables and update
 - definitions 163, 179
- working with the contents of logs 144
- working with the log data manager
 - option 205



Printed in USA

SH19-6816-14

