Contents

Figures ........................................ v

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
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 .. 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 DRLNJOB2 (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
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
eroutines .................................................. 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
Back 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 un installing a component ........ 126
Installing a component ................................ 126
Uninstalling a component ........................... 134
Working with table space profiles ............... 134

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 files ......................... 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 ..................... 211
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 DRLJLDML collect job and the parameters it uses .............................. 215
Deciding which log data sets to collect ..................................................... 215
Concatenation of log data sets ................................................................. 216
Running collect jobs in parallel .................................................................. 215
DRLJLDML sample job ........................................................................... 216
Setting the DYNAMNB value .................................................................. 218
Setting the parameters for job DRLJLDML .............................................. 218
Modifying the list of successfully collected log data sets ............................. 218
Viewing the information about successfully collected log data sets ............ 219
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
DRLREPORTS ....................................................................................... 230
DRLREPORT_ATTR .............................................................................. 238
DRLREPORT_COLUMN ....................................................................... 239
DRLREPORT_QUERY ............................................................................. 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

Contents V
Figures

1. Tivoli Decision Support for z/OS overview .................. 4
2. Overview of Tivoli Decision Support for z/OS data flow .......... 9
3. Administration window .................................. 10
4. Introducing the Reporting dialog ................................ 11
5. Tivoli Decision Support for z/OS Primary Menu ................. 30
6. System window - Option 1 .................................... 31
7. Dialog Parameters window .................................... 31
8. System window - Option 3 .................................... 33
9. System Tables (not created) window .......................... 34
10. Logs window .................................................. 37
11. Sample log statistics output .................................. 37
12. Reports window ............................................... 38
13. Data Selection window ....................................... 38
14. System Tables (created) window .............................. 51
15. Dialog Parameters window, when QMF is used .................. 57
16. Dialog Parameters window, when QMF is not used ............. 58
17. Tivoli Decision Support for z/OS definition member DRLISAMP, setting component definitions .................. 72
18. Tivoli Decision Support for z/OS definition member DRLLSAMP, defining a log type ...................... 74
19. Tivoli Decision Support for z/OS definition member DRLLSAMP, defining a record type ...................... 75
20. Tivoli Decision Support for z/OS definition member DRLLSAMP, defining a table space ...................... 75
21. Tivoli Decision Support for z/OS definition member DRLTSAMP, defining tables and updates (Part 1 of 2) ............ 76
22. Tivoli Decision Support for z/OS definition member DRLTSAMP, defining tables and updates (Part 2 of 2) ............ 77
23. Tivoli Decision Support for z/OS definition member DRLSAMP, defining reports and report groups ............ 79
24. Tivoli Decision Support for z/OS definition member DRLSAMP, report query .................................. 80
25. Invoking the log collector in batch to collect data ................ 86
26. DRLJCOLL job for collecting data from an SMF data set (Part 1 of 2) .................................. 87
27. DRLJCOLL job for collecting data from an SMF data set (Part 2 of 2) .................................. 88
28. Sample collect messages ...................................... 90
29. Collect Statistics window ...................................... 93
30. DB2 environment for the Tivoli Decision Support for z/OS database .................................. 96
31. Tablespace list window ..................................... 97
32. Tables window - Option 12 .................................. 101
33. Tablespace list window ..................................... 102
34. DRLJPURG job that uses all purge conditions .................. 104
35. Tables window - Option 10 .................................. 105
36. DRLJCOPY job for backing up Tivoli Decision Support for z/OS table spaces .................................. 106
37. DRLJRUNS job for generating DB2 statistics .................. 109
38. DB2I Primary Option Menu .................................. 110
39. Converting saved graphic report data to a page segment ............ 120
40. Using reports in BookMaster documents .................... 120
41. Using QMF to report in batch ................................ 120
42. Space pull-down ................................................ 127
43. Installation Options window .................................. 128
44. Sample log collector messages .................................. 129
45. Lookup Tables window ....................................... 130
46. Editing an installation job .................................... 131
47. Select Table window ......................................... 132
48. Tables window - showing component's lookup tables ............ 132
49. Component window ........................................... 137
50. Data Sets window ............................................. 145
51. Collect Statistics window ..................................... 145
52. Collect window ................................................. 147
53. Sample log statistics output .................................. 148
54. Record Data window .......................................... 149
55. List Record window ............................................ 150
56. Output from List record function ................................ 151
57. Log Definition window ....................................... 152
58. Record Definitions window .................................... 155
59. Record Definition window .................................... 156
60. Field Definition window ..................................... 157
61. Section Definition window ................................... 158
62. Record Procedure Definition window ......................... 161
63. Tables window ................................................. 164
64. Using QMF to display a Tivoli Decision Support for z/OS table .................................. 165
65. Editing a table in ISPF ....................................... 167
66. Table Size window ............................................ 168
67. Recalculate window .......................................... 169
68. Condition window ............................................ 170
69. Column Values window ....................................... 171
70. Selecting tables to unload .................................... 173
71. Unload Utility window ....................................... 174
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
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.</td>
<td>Tablespace window</td>
</tr>
<tr>
<td>91.</td>
<td>View window</td>
</tr>
<tr>
<td>92.</td>
<td>New Table window</td>
</tr>
<tr>
<td>93.</td>
<td>Grant Privilege window</td>
</tr>
<tr>
<td>94.</td>
<td>Revoke Privilege window</td>
</tr>
<tr>
<td>95.</td>
<td>Log Data Manager Main Selection window</td>
</tr>
<tr>
<td>96.</td>
<td>Collect Statements window</td>
</tr>
<tr>
<td>97.</td>
<td>Edit collect statements window</td>
</tr>
<tr>
<td>98.</td>
<td>Add Collect Statements Definition window</td>
</tr>
<tr>
<td>99.</td>
<td>Modify Collect Statements Definition window</td>
</tr>
<tr>
<td>100.</td>
<td>SMF Log Data Sets To Be Collected window</td>
</tr>
<tr>
<td>101.</td>
<td>Modify Log ID For a Log Data Set window</td>
</tr>
<tr>
<td>102.</td>
<td>Add a Data Set To Be Collected window</td>
</tr>
<tr>
<td>103.</td>
<td>Log Data Sets Collected Successfully window</td>
</tr>
<tr>
<td>104.</td>
<td>Retention Period window</td>
</tr>
<tr>
<td>105.</td>
<td>Log Data Sets Collected with Failure window</td>
</tr>
<tr>
<td>106.</td>
<td>Sample data flow</td>
</tr>
<tr>
<td>107.</td>
<td>Sample Report 1</td>
</tr>
<tr>
<td>108.</td>
<td>Sample Report 2</td>
</tr>
<tr>
<td>109.</td>
<td>Sample Report 3</td>
</tr>
<tr>
<td>110.</td>
<td>Part of an Indexspace Cross-reference report</td>
</tr>
<tr>
<td>111.</td>
<td>Part of an Actual Tablespace Allocation report</td>
</tr>
<tr>
<td>112.</td>
<td>Part of a Table Purge Condition report</td>
</tr>
<tr>
<td>113.</td>
<td>Example of List columns for a requested table with comment</td>
</tr>
<tr>
<td>114.</td>
<td>Example of List all tables with comment</td>
</tr>
<tr>
<td>115.</td>
<td>Example of REXX-SQL interface call</td>
</tr>
</tbody>
</table>
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:

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:


Ordering publications

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


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 graphic 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

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.
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,
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.
5. Create and run reports.

All of these options are available from the Administration window [Figure 3].
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.

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>
<thead>
<tr>
<th>KPM Component</th>
<th>Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Performance Metrics – z/OS</td>
<td>System Performance Feature Reference Volume I</td>
</tr>
<tr>
<td>Key Performance Metrics – CICS</td>
<td>CICS Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>Key Performance Metrics – DB2</td>
<td>System Performance Feature Reference Volume I</td>
</tr>
<tr>
<td>Key Performance Metrics - IMS</td>
<td>IMS Performance Feature Guide and Reference</td>
</tr>
</tbody>
</table>
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](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.
## Install 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:

<table>
<thead>
<tr>
<th>Program number</th>
<th>Product name and minimum VRM/service level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5625 – DB2</td>
<td>IBM DB2 Universal Database™ for z/OS Version 10</td>
</tr>
<tr>
<td>5615-DB2</td>
<td>IBM DB2 for z/OS Version 11</td>
</tr>
<tr>
<td>5694-A01</td>
<td>z/OS Version 1.13</td>
</tr>
<tr>
<td>5650-ZOS</td>
<td>z/OS Version 2.1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Product number</th>
<th>Product name and minimum VRM/service level</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>5625-DB2</td>
<td>Query Management Facility (QMF) for z/OS Version 8</td>
<td>Generate and view reports</td>
</tr>
<tr>
<td>5695-167</td>
<td>Graphical Data Display Manager (GDDM) Version 3.2</td>
<td>Display reports in graphical format</td>
</tr>
<tr>
<td>5668-812</td>
<td>GDDM – PGF Version 2.1.3</td>
<td>Transform reports into graphical format</td>
</tr>
<tr>
<td>5698-SD9</td>
<td>Tivoli Information Management for z/OS Version 7.1</td>
<td>Generate problem reports from Tivoli Decision Support for z/OS data</td>
</tr>
<tr>
<td>5722-SS1</td>
<td>OS/400® Version 5.1</td>
<td>AS/400 system performance</td>
</tr>
<tr>
<td>5685-108</td>
<td>NetView® FTP Version 2.1</td>
<td>AS/400 system performance</td>
</tr>
<tr>
<td>5733-196</td>
<td>NetView FTP/400 Version 3</td>
<td>AS/400 system performance</td>
</tr>
<tr>
<td>5655-AA1</td>
<td>DB2 High Performance Unload (HPU) Version 4.2</td>
<td>Unload DB2 data enhancement</td>
</tr>
</tbody>
</table>

Any one of the following:

<table>
<thead>
<tr>
<th>Program number</th>
<th>Product name and minimum VRM/service level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5765-E61</td>
<td>AIX 5L™ Version 5.1</td>
</tr>
<tr>
<td></td>
<td>HP – UX® Version 11-i</td>
</tr>
<tr>
<td></td>
<td>Sun Solaris Version 9</td>
</tr>
<tr>
<td></td>
<td>Linux RedHat Version 7.1 (Kernel 2.4.2)</td>
</tr>
<tr>
<td></td>
<td>Linux SUSE Version 7.1 (Kernel 2.4.0)</td>
</tr>
<tr>
<td></td>
<td>SLES 8 for zSeries</td>
</tr>
<tr>
<td></td>
<td>RedHat Enterprise Linux 3 for zSeries</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Program number</th>
<th>Product name and minimum VRM/service level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5694-A01</td>
<td>z/OS Version 1.13</td>
</tr>
</tbody>
</table>
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

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

Local data sets

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

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.

<table>
<thead>
<tr>
<th>Data set name</th>
<th>Description</th>
</tr>
</thead>
</table>

Chapter 2. Installing Tivoli Decision Support for z/OS  17
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.
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 DRLPROF, described in "Step 4: Preparing the dialog and updating the dialog profile” on page 27. Variables def_syspref, def_othbtpfx, def_iduser1, and def_idsq1user 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 SYSDM 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 "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

1. BIND the TDSz DB2 plan.
2. Create storage group and database for Tivoli Decision Support for z/OS.
3. Create views on the DB2 catalog.

Before you submit the job, do the following:

1. Check that the DB2 and TDSz data set names are correct. Search for db2loadlibrary and DRLvrm 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 of 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 DRLPROF, 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 occurrences 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.
11. If you are using a table prefix that is different from the default (DRL), change all occurrences 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

// ** OCCURRENCES 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 *
// ** OCCURRENCES 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 Kopied. *
** 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 - PO49985 :
** ADDED NOTE (POINT 11) IN ORDER TO *
** NOTIFY THAT, FOR MIGRATIONS, *
** EXISTING OBJECTS NEED BE DROPPED *
** BEFORE CREATE *
** 02 2002-05-27 BB - PO61494 :
** 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 PO49985 AND PO61494 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. *
** $D0=PK54663,TDS180,15/10/07,ADL(MG): Add CCSID EBCDIC to *
** CREATE DATABASE and Encoding*
** EBCDIC to BIND PLAN *
** $D1=DCR116, TDS181,15/05/09,ADL(RC): Update TDS Version *
** $D1=PM07437,TDS181,05/10/10,ADL(SY): Add comments for *
** using a different plan name *
** $D2=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=RW127524,TDS182,02/03/15,ADL(SY): Bind to package *
** ***********************************************************************************
** /DBINIT EXEC PGM=IKJEFT01 *
** /STEPXIB DD DISP=SHR,DSN=db2loadlib *
** /DBRMXIB DD DISP=SHR,DSN=DRLvrm.SDRLCNTL(DRDSQLX) *
** /SYSXPRINT DD SYSOUT=* *
** /SYSXSPRT DD SYSOUT=* *
** /SYSXTIN DD * *
** DSN SYSTEM(DSN) *
** BIND PACKAGE(DRLPLAN) OWNER(authid) MEMBER(DRDSQLX) -
** ACTION(REPLACE) ISOLATION(CS) ENCODING(EBCDIC) *
** BIND PLAN(DRLPLAN) OWNER(authid) PKLIST(*,DRLPLAN,*) -
** ACTION(REPLACE) RETAIN *
** RUN PROGRAM(DSNTIAD) PLAN(DSNTIAD) -

24 IBM Tivoli Decision Support for z/OS: Administration Guide and Reference
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'
//***************************************************************
//LICENSED MATERIALS - PROPERTY OF IBM
//5698-B06 Copyright IBM Corporation 1992, 2015
//SEE COPYRIGHT INSTRUCTIONS.
//***************************************************************
NAME: DRLJDBIP
STATUS: Tivoli Decision Support for zOS 1.8.2
FUNCTION: CREATE STORAGE GROUPS FOR PARTITIONING
NOTES: YOU NEED DB2 SYSADM AUTHORITY TO SUCCESSFULLY EXECUTE THIS JOB. BEFORE YOU SUBMIT THE JOB, DO THE FOLLOWING:
1. CHECK THAT THE DB2 AND TDSz DATA SET NAMES ARE CORRECT. SEARCH FOR db2loadlibrary AND DRLvrm TO FIND THE DATA SET NAMES.
2. IF THE DB2 SUBSYSTEM NAME IS NOT DSN, CHANGE SYSTEM=DSN TO SYSTEM=SUBSYSTEM-NAME IN THE SYSTSIN STEP.
3. IF YOU ARE USING A SYSTEM TABLE PREFIX THAT IS DIFFERENT FROM THE DEFAULT (DRLSYS), CHANGE DRLSYS IN THE GRANT STATEMENT TO THE NEW NAME.
4. IF YOU ARE USING A TABLE PREFIX THAT IS DIFFERENT FROM THE DEFAULT (DRL), CHANGE DRL IN THE GRANT STATEMENT TO THE NEW NAME.
5. IF YOU ARE USING STORAGE GROUP DIFFERENT FROM THE DEFAULT, CHANGE STOG1, STOG2, STOG3, STOG4, IN THE CREATE AND GRANT STATEMENTS. IF YOU NEED MORE/LESS STORAGE GROUPS MODIFY THE NUMBER (ADD/DELETE) OF CREATE AND GRANT STATEMENTS.
6. SPECIFY THE VOLUMES YOU ARE USING FOR THE STORAGE GROUPS (VOLSTGRxx).
7. SPECIFY THE VCAT PARAMETER IN THE CREATE STATEMENT (CATALOG-NAME)
8. IF YOU ARE NOT USING DB2 10.1, CHANGE DSNTIAB1 TO THE NAME OF THE CORRESPONDING PLAN FOR YOUR RELEASE.
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.
## Procedure

1. Make the load library (DRL182.SDRLLOAD), DB2 load library, QMF load library, and the GDDM load library accessible by performing one of these tasks:
   a. Allocate the SDRLLOAD 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.SDRLLOAD
   // DD DISP=SHR, DSN=QMF710.SDSQLOAD
   // DD DISP=SHR, DSN=GDDM.SADMMOD
   // DD DISP=SHR, DSN=DSN710.SDSNLOAD
   ```

   b. Add SDRLLOAD, SDSQLOAD, SADMMOD, and SDSNLOAD to the link list.

   c. Copy SDRLLOAD, 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 ALTLIB function to allocate the libraries.

      If Tivoli Decision Support for z/OS is invoked by using the ALTLIB 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 `ALTLIB 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.

2. If you start from the Primary Menu, type 2 Administration, 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: +
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.
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)
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 DRLJCSBTB 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 DRLJCSBTB.

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.
- ISPLIB, ISPLIB, ISPLIB, 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 stream data, or a data set, containing a command to invoke DRLEINIT, for example:
  IPSTART CMD(%DRLEINIT RESET)
- DRLBIN (batch input) DD referring to stream 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
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 table spaces 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).
3. From the Logs window, select the SAMPLE log and press F11.
   The Collect window is displayed.

4. Type DRLxxx.SDRLDEFS(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.

5. Press F3 to return to the Logs window after you finish looking at the
   messages.

6. Press F3 to return to the Administration window

7. From the Administration window, select 5, Reports, and press Enter.
The **Reporting Dialog Defaults** window is displayed. (Refer to *Guide to Reporting* for more information.)

8. 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 PRA000

**************************************************************************************
** 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**

9. 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  F3=Prompt  F4=Groups  F5=Table  F6=Chart  F7=Bkwd
F8=Fwd  F9=Swap  F10=Showfld  F11=Hdrval  F12=Cancel
```

**Figure 13. Data Selection window**

10. 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 Tablespace, 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, 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 SDRLDEFS 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:

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Book name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 Performance</td>
<td>AS/400 System Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>CICS Performance</td>
<td>CICS Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>Distributed Systems Performance</td>
<td>Distributed Systems Performance Feature Guide and Reference, SH19-4018</td>
</tr>
<tr>
<td>IMS Performance</td>
<td>IMS Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>Network Performance</td>
<td>Network Performance Feature Installation and Administration</td>
</tr>
<tr>
<td>System Performance</td>
<td>System Performance Feature Reference Volume I and II</td>
</tr>
</tbody>
</table>

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.
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.


   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...

**DRL**DRL

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 “??????”
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 references 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.DAIL 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*.

**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>
<thead>
<tr>
<th>Table 2. Explanation of Program DRLCDATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input/output</strong></td>
</tr>
<tr>
<td>INPUT</td>
</tr>
<tr>
<td>INPUT</td>
</tr>
</tbody>
</table>
Table 2. Explanation of Program DRLCADATA (continued)

<table>
<thead>
<tr>
<th>Input/output</th>
<th>DDNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT</td>
<td>CIMSSMF</td>
<td>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.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSACCT</td>
<td>This data set contains selected SMF chargeback records (6, 30, 101, 110). This data set is used as input in step DRLC2B.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSCICS</td>
<td>This data set contains CICS records (SMF Type 110). This record is used by the Usage and Accounting Collector CICS interface programs.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSDB2</td>
<td>This data set contains DB2 records (SMF Type 101). This record is used by the Usage and Accounting Collector DB2 interface programs.</td>
</tr>
</tbody>
</table>

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 DRLNSMF 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 CIMSACT) 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

<table>
<thead>
<tr>
<th>Input/output</th>
<th>DDNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>CIMSDATA</td>
<td>Reformatted SMF records. These records are created by DDNAME CIMSACT in program DRLCDATA. The Usage and Accounting Collector Suspense file for unmatched job step and print records is appended to DDNAME CIMSDATA.</td>
</tr>
<tr>
<td>INPUT</td>
<td>CIMSCNTL</td>
<td>Control statements.</td>
</tr>
</tbody>
</table>
Table 3. Explanation of Program DRLCACCT (continued)

<table>
<thead>
<tr>
<th>Input/output</th>
<th>DDNAME</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>CIMSTABL</td>
<td>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 Usage and Accounting Collector User Guide.</td>
</tr>
<tr>
<td>INPUT</td>
<td>CIMSDTVS</td>
<td>Usage and Accounting Collector Dictionary VSAM file.</td>
</tr>
<tr>
<td>INPUT</td>
<td>CIMSPDS</td>
<td>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.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>IMSACT2</td>
<td>Usage and Accounting Collector batch chargeback file containing the 79x accounting records. This data set is used by DRLCEXTR and DRLCMONY.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSUSPN</td>
<td>Suspense file. This data set contains Step and Print records that have not been matched with a Job Start or Job Stop record.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSEXCP</td>
<td>This data set contains records that have not been matched with entries in the CIMSTABL data set.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSPRNT</td>
<td>This data set contains the runtime parameters and the results of the run.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSMSG</td>
<td>This data set contains informational messages.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSSEL</td>
<td>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.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>CIMSUNSP</td>
<td>Unsupported CSR records. This data set contains all CSR records that did not have a definition within CIMSDTVS.</td>
</tr>
</tbody>
</table>

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).
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.

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

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>
<thead>
<tr>
<th>Subsystem Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLNCICS</td>
<td>CICS Support</td>
</tr>
<tr>
<td>DRLNDDB2</td>
<td>DB2</td>
</tr>
<tr>
<td>DRLNMQSRR</td>
<td>MQSeries®</td>
</tr>
<tr>
<td>DRLNDISK</td>
<td>DASD Space</td>
</tr>
<tr>
<td>DRLNTAPE</td>
<td>Tape Storage</td>
</tr>
<tr>
<td>DRLNIMS</td>
<td>IMS</td>
</tr>
<tr>
<td>DRLNUNIV</td>
<td>ROSCOE, ADABAS/SME, IDMS/SME, RJE, WYLBUR, Oracle, MEMO, Control-T, BETA</td>
</tr>
</tbody>
</table>

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).
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.
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.
Dialog Parameters

Type information. Then press Enter to save and return.

More: +

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 subsystem name</td>
<td>DSN</td>
</tr>
<tr>
<td>DB2 plan name for TDS</td>
<td>DRLPLAN</td>
</tr>
<tr>
<td>Database name</td>
<td>DRLDB</td>
</tr>
<tr>
<td>Storage group default</td>
<td>DRLSG</td>
</tr>
<tr>
<td>Prefix for system tables</td>
<td>DRLSYS</td>
</tr>
<tr>
<td>Prefix for all other tables</td>
<td>DRL</td>
</tr>
<tr>
<td>Show TDS environment data</td>
<td>NO or YES</td>
</tr>
<tr>
<td>Buffer pool for data</td>
<td>BPO</td>
</tr>
<tr>
<td>Buffer pool for indexes</td>
<td>BPO</td>
</tr>
<tr>
<td>Users to grant access to</td>
<td>DRLUSER</td>
</tr>
<tr>
<td>SQL ID to use (in QMF)</td>
<td>DRLUSER</td>
</tr>
<tr>
<td>QMF language</td>
<td>PROMPTED (SQL or PROMPTED)</td>
</tr>
<tr>
<td>SYSOUT class (in QMF)</td>
<td>Q</td>
</tr>
<tr>
<td>Default printer</td>
<td></td>
</tr>
<tr>
<td>SQLMAX value</td>
<td>5000</td>
</tr>
<tr>
<td>Reporting dialog mode</td>
<td>1. End user mode</td>
</tr>
<tr>
<td></td>
<td>2. Administrator mode</td>
</tr>
<tr>
<td>Dialog language</td>
<td>1. English</td>
</tr>
<tr>
<td></td>
<td>2. Japanese</td>
</tr>
<tr>
<td>DB2 data sets</td>
<td></td>
</tr>
<tr>
<td>Prefix</td>
<td>DSN810</td>
</tr>
<tr>
<td>Suffix</td>
<td></td>
</tr>
<tr>
<td>QMF data sets prefix</td>
<td>QMF810</td>
</tr>
<tr>
<td>Tivoli Decision Support for z/OS data sets prefix</td>
<td>DRL181</td>
</tr>
<tr>
<td>Temporary data sets prefix</td>
<td>(user_ID substituted)</td>
</tr>
<tr>
<td>Local definitions data set</td>
<td>DRL.LOCAL.DEFS</td>
</tr>
<tr>
<td>Local GDDM formats data set</td>
<td>DRL.LOCAL.ADMCFORM</td>
</tr>
<tr>
<td>Local messages data set</td>
<td>DRL.LOCAL.MESSAGES</td>
</tr>
<tr>
<td>Saved reports data set</td>
<td>DRL.LOCAL.REPORTS</td>
</tr>
<tr>
<td>Saved charts data set</td>
<td>DRL.LOCAL.CHARTS</td>
</tr>
<tr>
<td>Job statement information</td>
<td></td>
</tr>
<tr>
<td>(required for batch jobs):</td>
<td></td>
</tr>
<tr>
<td>// (user_ID substituted)</td>
<td>JOB (000000,XXXX),USER1,MSGLEVEL=(1,1),</td>
</tr>
<tr>
<td>// NOTIFY=(user_ID substituted),MSGCLASS=Q,CLASS=E,REGION=4096K</td>
<td></td>
</tr>
<tr>
<td>/*</td>
<td></td>
</tr>
<tr>
<td>F1=Help</td>
<td></td>
</tr>
<tr>
<td>F2=Split</td>
<td></td>
</tr>
<tr>
<td>F7=Bkwd</td>
<td></td>
</tr>
<tr>
<td>F8=Fwd</td>
<td></td>
</tr>
<tr>
<td>F9=Swap</td>
<td></td>
</tr>
<tr>
<td>F12=Cancel</td>
<td></td>
</tr>
</tbody>
</table>

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

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

Table:

<table>
<thead>
<tr>
<th>Variable/Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 subsystem name</td>
<td>DSN</td>
</tr>
<tr>
<td>DB2 plan name for TDS</td>
<td>DRLPLAN</td>
</tr>
<tr>
<td>Database name</td>
<td>DRLDB</td>
</tr>
<tr>
<td>Storage group default</td>
<td>DRLSG</td>
</tr>
<tr>
<td>Prefix for system tables</td>
<td>DRLSYS</td>
</tr>
<tr>
<td>Prefix for all other tables</td>
<td>DRL</td>
</tr>
<tr>
<td>Show TDS environment data</td>
<td>NO (YES or NO)</td>
</tr>
<tr>
<td>Buffer pool for data</td>
<td>BP0</td>
</tr>
<tr>
<td>Buffer pool for indexes</td>
<td>BP0</td>
</tr>
<tr>
<td>Users to grant access to</td>
<td>DRLUSER</td>
</tr>
<tr>
<td>Batch print SYSOUT class</td>
<td>A</td>
</tr>
<tr>
<td>Printer line count per page</td>
<td>60</td>
</tr>
<tr>
<td>SQLMAX value</td>
<td>5000</td>
</tr>
<tr>
<td>Reporting dialog mode</td>
<td>1. End user mode, 2. Administrator mode</td>
</tr>
<tr>
<td>Dialog language</td>
<td>1. English, 2. Japanese</td>
</tr>
<tr>
<td>DB2 data sets</td>
<td>DB2.V820</td>
</tr>
<tr>
<td>TDS for zOS data sets prefix</td>
<td>DRL182</td>
</tr>
<tr>
<td>Temporary data sets prefix</td>
<td>(user_ID substituted)</td>
</tr>
<tr>
<td>Local defs data set</td>
<td>DRL.LOCAL.DEFS</td>
</tr>
<tr>
<td>Local User defs data set</td>
<td>DRL.LOCAL.USER.DEFS</td>
</tr>
<tr>
<td>Local GDDM formats data set</td>
<td>DRL.LOCAL.ADMCFORM</td>
</tr>
<tr>
<td>Local messages data set</td>
<td>DRL.LOCAL.MESSAGES</td>
</tr>
<tr>
<td>Saved reports data set</td>
<td>DRL.LOCAL.REPORTS</td>
</tr>
<tr>
<td>Saved charts data set</td>
<td>DRL.LOCAL.CHARTS</td>
</tr>
<tr>
<td>Job statement information (required for batch jobs):</td>
<td></td>
</tr>
<tr>
<td>// (user_ID substituted) JOB (000000,XXXX), 'USER1', MSGLEVEL(1,1),</td>
<td></td>
</tr>
<tr>
<td>// NOTIFY=ASYSUID, MSGCLASS=Q, CLASS=E, REGION=4096K</td>
<td></td>
</tr>
<tr>
<td>F1=Help, F2=Split, F7=Bkwd, F8=Fwd, F9=Swap, F12=Cancel</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>userid.DRLFPROF variable name</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>modtenu</td>
<td>N/A</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>The fully qualified name of the user tables’ library, if any. The maximum supported value is 99999999.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2plib2</td>
<td>N/A</td>
<td>SDSNPP</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2plibe</td>
<td>N/A</td>
<td>SDSNPPPPE</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2plibk</td>
<td>N/A</td>
<td>SDSNPPPK</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmprint</td>
<td>N/A</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Specifies whether the QMF output is saved in the DSQPRINT data set (YES) or in the SYSOUT class (NO).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_db2subs</td>
<td>DB2 subsystem name</td>
<td>DSN</td>
<td></td>
</tr>
<tr>
<td>The DB2 subsystem where Tivoli Decision Support for z/OS resides.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be 4 alphanumeric characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_db2plan</td>
<td>DB2 plan name for TDS</td>
<td>DRLPLAN</td>
<td></td>
</tr>
<tr>
<td>The DB2 plan name to which the distributed Tivoli Decision Support for z/OS for z/OS DBRM has been bound.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be 8 alphanumeric characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Only change the plan name shown here if instructed to do so by your Tivoli Decision Support for zOS system administrator.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_dbname</td>
<td>Database name</td>
<td>DRLDB</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default database is DRLDB. If this value is something other than DRLDB, it is likely the default value for your site.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_storgrp</td>
<td>Storage group default</td>
<td>DRLSG</td>
<td></td>
</tr>
<tr>
<td>The storage group that Tivoli Decision Support for z/OS uses for the DB2 database identified in the Database name field.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be 8 alphanumeric characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default is DRLSG. If the value of the field is something other than DRLSG, it was changed during installation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_syspref</td>
<td>Prefix for system tables</td>
<td>DRLSYS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Dialog parameters - variables and fields

<table>
<thead>
<tr>
<th>userid.DRLFPROF variable name</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>Prefixed system and control DB2 tables.</td>
<td>DRLSYS</td>
<td>Your value</td>
</tr>
<tr>
<td>This required field can be 8 alphanumerical characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default is DRLSYS. If the value is something other than DRLSYS, it was changed during installation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do not change the value; Tivoli Decision Support for z/OS uses this value to access its system tables.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_othtpfx</td>
<td>Prefix for all other tables</td>
<td>DRL</td>
<td></td>
</tr>
<tr>
<td>The prefix of Tivoli Decision Support for z/OS data tables in the DB2 database.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid values are determined at installation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be 8 alphanumerical characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default is DRL. If the value is something other than DRL, it was changed during installation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_drlshwid</td>
<td>Show TDS environment data</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Specifies whether or not to display the Tivoli Decision Support for z/OS environment data in the main panels.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can have a value of YES or NO.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default value for this field is NO.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_tsbspool</td>
<td>Buffer pool for data</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_ixbspool</td>
<td>Buffer pool for indexes</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>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).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_iduser1</td>
<td>Users to grant access to</td>
<td>DRLUSER</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You must specify a value for at least one of the fields.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Each user ID or group ID can be 8 alphanumerical characters. The first character must not be numeric.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You can grant users access to the tables and views of a component by listing them here before you install the component.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_idsqlusr</td>
<td>SQL ID to use (in QMF)</td>
<td>DRLUSER</td>
<td></td>
</tr>
<tr>
<td>This field is used only if your installation uses QMF.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be up to 8 alphanumerical characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default is DRLUSER. If the value is something other than DRLUSER, it was changed during installation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You can change this value to your user ID if you have DB2 authorization to Tivoli Decision Support for z/OS tables and views.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### userid.DRLFPROF variable name

<table>
<thead>
<tr>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>def_qmflng</td>
<td>QMF language</td>
<td>PROMPTED</td>
</tr>
</tbody>
</table>

The QMF language for creating reports and queries, either SQL (structured query language) or PROMPTED QUERY.

PROMPTED QUERY is the default QMF language for Tivoli Decision Support for z/OS.

This is a required field, if your installation uses QMF.

<table>
<thead>
<tr>
<th>def_qmfprt</th>
<th>SYSOUT class (in QMF)</th>
<th>Q</th>
</tr>
</thead>
</table>

The SYSOUT class for report data sets that QMF generates, or for output that QMF routes to a printer. The default value is Q.

This is a required field, if your installation uses QMF.

<table>
<thead>
<tr>
<th>def_printer</th>
<th>Default printer</th>
<th>blank</th>
</tr>
</thead>
</table>

The GDDM nickname of a printer to use for printing graphic reports. The printer should be one capable of printing GDDM-based graphics.

The printer name must be defined in the GDDM nicknames file, allocated to the ADMDEFS ddname. Refer to QMF: Reference and GDDM User’s Guide for more information about defining GDDM nicknames.

<table>
<thead>
<tr>
<th>def_dr1prt</th>
<th>Batch print SYSOUT class</th>
<th>A</th>
</tr>
</thead>
</table>

This field is used only if your installation does not use QMF.

A valid SYSOUT class for printing tabular reports in batch. Valid values are A-Z, 0-9, and *.

<table>
<thead>
<tr>
<th>def_pagelen</th>
<th>Printer line count per page</th>
<th>60</th>
</tr>
</thead>
</table>

This field is used only if your installation does not use QMF.

The number of report lines that should be printed on each page when you print tabular reports online and in batch.

<table>
<thead>
<tr>
<th>def_drlmax</th>
<th>SQLMAX value</th>
<th>5000</th>
</tr>
</thead>
</table>

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.

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.

<table>
<thead>
<tr>
<th>def_rptdialg</th>
<th>Reporting dialog mode</th>
<th>1</th>
</tr>
</thead>
</table>

The dialog mode for using the reporting dialog. Any option you save applies to future sessions.

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).

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.

<table>
<thead>
<tr>
<th>N/A</th>
<th>Dialog language</th>
<th>1</th>
</tr>
</thead>
</table>

The language in which Tivoli Decision Support for z/OS displays all its windows.

Tivoli Decision Support for z/OS supports those languages listed in the window. Choose the language your site has installed.

If you leave this field blank, Tivoli Decision Support for z/OS displays its windows in English.

Any changes you make to this field become effective in your next dialog session, when Tivoli Decision Support for z/OS allocates its libraries.

<table>
<thead>
<tr>
<th>def_db2dspfx</th>
<th>DB2 data sets-prefix</th>
<th>DSN710</th>
</tr>
</thead>
</table>
## Dialog parameters - variables and fields

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Default Value</th>
<th>Your Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>userid.DRLFPROF</td>
<td>The prefix to which Tivoli Decision Support for z/OS appends DB2 data set names as it performs tasks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field is required if db2def is SUFFIX. If db2def is DATASET, this field is ignored.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field can be 35 alphanumeric characters.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The default is DB2.V810. If the value of this field is something other than DB2.V810, it was changed during installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2 data sets-suffix</td>
<td>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.</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field can be used if db2def is SUFFIX. If db2def is DATASET, this field is ignored.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field can be 35 alphanumeric characters.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Do not use this field to qualify data sets that you create; this is not its purpose. Use it to identify DB2 modules only.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_qmfdspfx</td>
<td>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.</td>
<td>QMF710</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If qmfddef is SUFFIX, this field is required. If qmfddef is DATASET, this field is ignored.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The default is DB2.V810. If the value is something other than DB2.V810, it was changed during installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Any changes you make to this field become effective in your next session, when Tivoli Decision Support for z/OS allocates its libraries.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_dsnpref</td>
<td>The prefix of Tivoli Decision Support for z/OS libraries.</td>
<td>DRL182</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This required field can be up to 35 alphanumeric characters.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The default is DRL182. If the value of this field is something other than DRL182, it was changed during installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Any changes you make to this field become effective in your next session, when Tivoli Decision Support for z/OS allocates its libraries.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No equivalent</td>
<td>Temporary data sets prefix</td>
<td>user_ID</td>
<td></td>
</tr>
<tr>
<td>userid.DRLPROF variable name</td>
<td>Dialog Parameters field name</td>
<td>Default value</td>
<td>Your value</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>The prefix for any temporary data sets you create while using Tivoli Decision Support for z/OS.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be up to 35 alphanumeric characters.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default value is your user_ID or the TSO_prefix.user_ID.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_dsnlocdn</td>
<td>Local definitions data set</td>
<td>DRL.LOCAL.DEFS</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This optional field can be 44 alphanumeric characters.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_usrlocdn</td>
<td>Local User alter/definitions data set</td>
<td>DRL.LOCAL.USER.DEFS</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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 user definition data sets.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_modform</td>
<td>The local GDDM formats data set</td>
<td>DRL.LOCAL.ADMCFORM</td>
<td></td>
</tr>
<tr>
<td>The data set where you keep your GDDM formats for graphic reports.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_drlmsgs</td>
<td>Local messages data set</td>
<td>DRL.LOCAL.MESSAGES</td>
<td></td>
</tr>
<tr>
<td>Use this field to identify a PDS that contains messages generated by users during communication with Tivoli Decision Support for z/OS administrators.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The value of this field depends on naming conventions that your Tivoli Decision Support for z/OS administrator has established.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This required field can be up to 44 alphanumeric characters.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_dsnrept</td>
<td>Saved reports data set</td>
<td>DRL.LOCAL.REPORTS</td>
<td></td>
</tr>
</tbody>
</table>
## Dialog parameters - variables and fields

<table>
<thead>
<tr>
<th>userid.DRLFPROF variable name</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The PDS where Tivoli Decision Support for z/OS saves your tabular reports.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This optional field can be up to 44 alphanumeric characters.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default PDS is DRL.LOCAL.REPORTS.</td>
<td>def_dsnchrts</td>
<td>Saved charts data set</td>
<td>DRL.LOCAL.CHARTS</td>
</tr>
<tr>
<td>The PDS where Tivoli Decision Support for z/OS saves the graphic reports you choose to save.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This optional field can be up to 44 alphanumeric characters.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default PDS is DRL.LOCAL.ADMGDF.</td>
<td>def_jclsta1, def_jclsta2, def_jclsta3, def_jclsta4</td>
<td>Job statement information (required for batch jobs)</td>
<td>Sample job card in which Tivoli Decision Support for z/OS dynamically substitutes the user ID.</td>
</tr>
<tr>
<td>The job statement information to be used for batch jobs that the dialogs create for you.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You must use correct JCL in the job statement. Tivoli Decision Support for z/OS does not validate job statement information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do not use JCL comments in these JCL statements.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You can specify up to four card images in these job statement fields.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The first &quot;//&quot; card image should contain the job name. Press Enter to save any job statements for all future sessions.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bkmgr_mlib</td>
<td>N/A</td>
<td>EOY.SEOMENU</td>
<td></td>
</tr>
<tr>
<td>The BookManager message library.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bkmgr_plib</td>
<td>N/A</td>
<td>EOY.SEOPENU</td>
<td></td>
</tr>
<tr>
<td>The BookManager panel library.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bkmgr_tlrib</td>
<td>N/A</td>
<td>EOY.SEOTENU</td>
<td></td>
</tr>
<tr>
<td>The BookManager tables library.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dr1sh00</td>
<td>N/A</td>
<td>IBMBK.DRLISH00. BKSHELF</td>
<td></td>
</tr>
<tr>
<td>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 DRLISH00 because you have installed a newer version of the online books and bookshelf.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If you do not plan to use BookManager, change the value of this field to blank.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsnnsufx</td>
<td>N/A</td>
<td>SDRL.DEFS</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS definitions data set suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>execsfx</td>
<td>N/A</td>
<td>SDRLEXEC</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS exec data set suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>loadsf</td>
<td>N/A</td>
<td>SDRL.LOAD</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS load library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>skelsfx</td>
<td>N/A</td>
<td>SDRL.SKEL</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS skeleton data set suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Dialog parameters - variables and fields

<table>
<thead>
<tr>
<th>user_id.DRLFPROF variable name</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>eng_lib_sfx</td>
<td>N/A</td>
<td>ENU</td>
<td></td>
</tr>
<tr>
<td>The English library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jpn_lib_sfx</td>
<td>N/A</td>
<td>JPN</td>
<td></td>
</tr>
<tr>
<td>The Japanese library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_nlslang</td>
<td>N/A</td>
<td>eng_lib_sfx</td>
<td></td>
</tr>
<tr>
<td>The national language library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>repsufx</td>
<td>N/A</td>
<td>“SDRLR”+def_nlslang</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS report definitions library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>plibsfx</td>
<td>N/A</td>
<td>“SDRLP”+def_nlslang</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS panel library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>messsfx</td>
<td>N/A</td>
<td>“SDRLM”+def_nlslang</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS message library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>formsfx</td>
<td>N/A</td>
<td>“SDRLF”+def_nlslang</td>
<td></td>
</tr>
<tr>
<td>The Tivoli Decision Support for z/OS GDDM formats library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eng_qmf_sfx</td>
<td>N/A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>The English library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jpn_qmf_sfx</td>
<td>N/A</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>The Kanji-Japanese library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_qmflang</td>
<td>N/A</td>
<td>eng_qmf_sfx</td>
<td></td>
</tr>
<tr>
<td>The national language default library suffix.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmfdef</td>
<td>N/A</td>
<td>SUFFIX</td>
<td></td>
</tr>
<tr>
<td>The method of describing QMF library names to Tivoli Decision Support for z/OS, either SUFFIX or DATASET.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In either case, Tivoli Decision Support for z/OS uses the next several variables to allocate QMF libraries.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmflib</td>
<td>N/A</td>
<td>SDSQCLST+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmfclibe</td>
<td>N/A</td>
<td>SDSQCLST+eng_qmf_sfx</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmfelib</td>
<td>N/A</td>
<td>SDSQEXEC+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmfelibe</td>
<td>N/A</td>
<td>SDSQEXEC+eng_qmf_sfx</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Chapter 3. Dialog parameters** 65
### Dialog parameters - variables and fields

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>userid.DRLFPROF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmfplib</td>
<td>N/A</td>
<td>SDSQPLIB+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>qmflib</td>
<td>N/A</td>
<td>SDSQPLIB+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>qmfslib</td>
<td>N/A</td>
<td>SDSQSLIB+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>qmfmmap</td>
<td>N/A</td>
<td>SDSQMAP+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>qmfpmn</td>
<td>N/A</td>
<td>DSQPNL+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>dsqpmn</td>
<td>N/A</td>
<td>DSQPNL+def_qmflang</td>
<td></td>
</tr>
<tr>
<td>qmflload</td>
<td>N/A</td>
<td>SDSQLOAD</td>
<td></td>
</tr>
<tr>
<td>qmfschart</td>
<td>N/A</td>
<td>DSQCHART</td>
<td></td>
</tr>
<tr>
<td>qmfdsdum</td>
<td>N/A</td>
<td>DUMMY</td>
<td></td>
</tr>
<tr>
<td>qmfdebug</td>
<td>N/A</td>
<td>DUMMY</td>
<td></td>
</tr>
<tr>
<td>dsunit</td>
<td>N/A</td>
<td>SYSDA</td>
<td></td>
</tr>
<tr>
<td>db2ver</td>
<td>N/A</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>db2rel</td>
<td>N/A</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>db2def</td>
<td>N/A</td>
<td>SUFFIX</td>
<td></td>
</tr>
</tbody>
</table>

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.

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.

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.

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.

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.

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.

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.

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.

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.

The fully-qualified name of the data set to be allocated to ddname DSQUDUMP, or DUMMY.

The fully-qualified name of the data set to be allocated to ddname DSQDEBUG, or DUMMY.

The disk unit.

The version of DB2. Must be a decimal number 1-99.

The release of DB2.
**Dialog parameters - variables and fields**

<table>
<thead>
<tr>
<th>userid.DRLFPROM variable name</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The method of describing DB2 library names to Tivoli Decision Support for z/OS, either SUFFIX or DATASET.</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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_db2dspx), a library name, and an optional suffix (def_db2dssfx).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In either case, Tivoli Decision Support for z/OS uses the next several variables to allocate DB2 libraries.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2lib</td>
<td>N/A</td>
<td>RUNLIB.LOAD</td>
<td></td>
</tr>
<tr>
<td>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_db2dspx before appending def_db2dssfx.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2load</td>
<td>N/A</td>
<td>SDSNLOAD</td>
<td></td>
</tr>
<tr>
<td>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_db2dspx before appending def_db2dssfx.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2clst</td>
<td>N/A</td>
<td>SDSNCLIST</td>
<td></td>
</tr>
<tr>
<td>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_db2dspx before appending def_db2dssfx.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2mlib</td>
<td>N/A</td>
<td>SDSNSPFM</td>
<td></td>
</tr>
<tr>
<td>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_db2dspx before appending def_db2dssfx.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db2plib</td>
<td>N/A</td>
<td>SDSNSPPF</td>
<td></td>
</tr>
<tr>
<td>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_db2dspx before appending def_db2dssfx.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gddmload</td>
<td>N/A</td>
<td>GDDM.SADMMOD</td>
<td></td>
</tr>
<tr>
<td>The GDDM load library.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admssymb</td>
<td>N/A</td>
<td>GDDM.SADMSYM</td>
<td></td>
</tr>
<tr>
<td>The GDDM symbols library.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admdefs</td>
<td>N/A</td>
<td>SYS1.GDDMNICK</td>
<td></td>
</tr>
<tr>
<td>The GDDM nicknames library.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admprntq</td>
<td>N/A</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_geapplid</td>
<td>N/A</td>
<td>zuser</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_gesessn</td>
<td>N/A</td>
<td>BLGSES00</td>
<td></td>
</tr>
<tr>
<td>The session member (module) used to start an Information/Management session.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_geprivcl</td>
<td>N/A</td>
<td>MASTER</td>
<td></td>
</tr>
<tr>
<td>The privilege class specified in an Information/Management group record.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIEWER</td>
<td>N/A</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>
# Dialog parameters - variables and fields

<table>
<thead>
<tr>
<th>userid.DRLFPROF variable name</th>
<th>Dialog Parameters field name</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qmfuse</td>
<td>N/A</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gddmuse</td>
<td>N/A</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>decsep</td>
<td>N/A</td>
<td>PERIOD</td>
<td></td>
</tr>
<tr>
<td>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=&quot;COMMA&quot;. 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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subhdrv</td>
<td>N/A</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_useaot</td>
<td>N/A</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Specifies whether Analytics component tables are created as Accelerator Only Tables in IBM DB2 Analytics Accelerator or as tables in DB2. &quot;YES&quot;: Tables are created as Accelerator Only Tables. &quot;NO&quot;: Tables are created in DB2 and are applicable for use either as DB2 tables or as IDAA_ONLY table.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_accelerator</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The name of the Accelerator where the Analytics components tables reside. Required only if using Accelerator Only Tables, that is, if def_useaot is set to &quot;YES&quot;.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def_timeint</td>
<td>N/A</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Specifies the time interval granularity for records collected for Analytics components tables. &quot;H&quot;: The timestamp for records is rounded to hourly intervals, which is similar to non-Analytics tables with a suffix of &quot;_H&quot; in other components. &quot;S&quot;: 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. &quot;T&quot;: The timestamp for tables is the actual timestamp in the SMF log record, which is similar to non-Analytics tables with suffix &quot;_T&quot;.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Replacing empty variables increases the time taken to generate a report.

Specify N to leave the empty variable in the report.
## Allocation overview

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

<table>
<thead>
<tr>
<th>Library type or data set ddname</th>
<th>Library or data set</th>
<th>Allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(EPDM exec)</td>
</tr>
<tr>
<td><strong>Tivoli Decision Support for z/OS allocates the following libraries as a user starts a Tivoli Decision Support for z/OS dialog:</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| ISPPLIB | • Tivoli Decision Support for z/OS panel library  
• QMF panel library  
• DB2 panel library | DRLEINI1 |
| ISPTLIB | • Tivoli Decision Support for z/OS tables library  
• QMF tables library  
• BookManager tables library | DRLEINI1 |
| ISPMLIB | • Tivoli Decision Support for z/OS message library  
• QMF message library  
• DB2 message library | DRLEINI1 |
| ISPLLIB | • Tivoli Decision Support for z/OS load library  
• QMF load library | DRLEINI1 |
| ISPSLIB | • 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_qmfdspx.qmflib+E) | DRLEQMF |
| SYSEXEC | QMF exec library (def_qmfdspx.qmfelib+E) | DRLEQMF |
| ADMGMAP | SDSQMAP library (def_qmfdspx.qmffmap) | DRLEQMF |
| ADMCFORM | Saved forms data set + DSQCHART library (dsnpref.formsfx + def_qmfdspx.qmffchart) | DRLEQMF |
| DSQUCFRM | Saved forms data set | DRLEQMF |
| DSQPNLE | QMF panel library | DRLEQMF |
| DSQPRINT | QMF sysout class (qmfprt) | DRLEQMF |
| DSQSPILL | NEW DELETE (temporary file allocation) | DRLEQMF |
| DSQEDIT | NEW DELETE (temporary file allocation) | DRLEQMF |
| DSQDEBUG | (qmfdebug) | DRLEQMF |
| DSQDUMP | (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

<table>
<thead>
<tr>
<th>Library type or data set ddname</th>
<th>Library or data set</th>
<th>Allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPROC</td>
<td>DB2 CLIST library (db2dspfx.db2clst)</td>
<td>DRLEDDB2I</td>
</tr>
</tbody>
</table>

Tivoli Decision Support for z/OS allocates the following library as a user starts DB2 Interactive (DB2I) from Tivoli Decision Support for z/OS:
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 DRLIxxxx members, in the DRL182.SDRLDEFS library, contain SQL statements that are executed.

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

```sql
/* 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', 'DRLSAMP');
```

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

<table>
<thead>
<tr>
<th>Order</th>
<th>Member naming convention</th>
<th>Definition types</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DRLlxxxx</td>
<td>Logs.</td>
</tr>
<tr>
<td>2</td>
<td>DRLRxxxx</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>DRLSxxxx</td>
<td>Tablespaces.</td>
</tr>
<tr>
<td>4</td>
<td>DRLTxxxx</td>
<td>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.</td>
</tr>
<tr>
<td>5</td>
<td>DRLUxxxx</td>
<td>Updates (also found in DRLTxxxx members).</td>
</tr>
<tr>
<td>6</td>
<td>DRLVxxxx</td>
<td>Views (also found in DRLTxxxx members).</td>
</tr>
<tr>
<td>7</td>
<td>DRLOxxxx</td>
<td>Report groups and reports. Report groups must be defined before the report definitions that reference them.</td>
</tr>
</tbody>
</table>

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

DRLLxxxx 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.

```sql
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.)
Defining table spaces

DRLSxxx 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 DRLSxxx 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.

```
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 tables and updates

DRLTxxx 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, DRLTSAEMP. 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**

```sql
/* Define table SAMPLE_USER */

CREATE TABLE &PREFIX.SAMPLE_USER
(USER_ID CHAR(8) NOT NULL,
 DEPARTMENT_NAME CHAR(8) NOT NULL,
 PRIMARY KEY (USER_ID))
IN &DATABASE.DRLSSAMP;

CREATE UNIQUE INDEX &PREFIX.SAMPUSER_IX
ON &PREFIX.SAMPLE_USER
(USER_ID)
USING STOGROUP &STOGROUP.
PRIQTY 12
SEQQTY 4
CLUSTER BUFFERPOOL &IXBUFFERPOOL;

/* Define comments for SAMPLE_USER */

COMMENT ON TABLE &PREFIX.SAMPLE_USER
IS 'This lookup table assigns department names to users.';

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

/* Grant users read access to SAMPLE_USER */

GRANT SELECT ON &PREFIX.SAMPLE_USER TO &USERS.;

/* Insert data in SAMPLE_USER */

INSERT INTO &PREFIX.SAMPLE_USER
VALUES('ADAMS ','Appl Dev');

/* Define table SAMPLE_H */

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)
Defining updates and views

DRLUXxxx members of the DRL182.SDRLDEFS library define updates not previously defined in DRLTxxx definition members. For example, member DRLUMAV in the DRL182.SDRLDEFS library defines updates from record types SMF_030 and SMF_070 to the AVAILABILITY_T table.

DRLVxxx members of the DRL182.SDRLDEFS library define views not previously defined in DRLTxxx definition members. For example, member DRLVC901 in the DRL182.SDRLDEFS 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

DRLOxxxx 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.
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:

<table>
<thead>
<tr>
<th>Member type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLQxxxx</td>
<td>Report queries in DRL182.SDRLRxxx</td>
</tr>
<tr>
<td>DRLFxxxx</td>
<td>Report forms in DRL182.SDRLRxxx</td>
</tr>
<tr>
<td>DRLGxxxx</td>
<td>Report charts in DRL182.SDRLFxxx</td>
</tr>
</tbody>
</table>

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.

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

DEFINE REPORT SAMPLE01
  VERSION 'IBM.110'
  DESC 'Sample Report 1'
  QUERY DRLQSA01
  FORM DRLFSA01
  CHART DRLGSRUF
  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 DRLGHRFB
  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.
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.

```sql
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
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:

<table>
<thead>
<tr>
<th>Naming convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLBxxxx</td>
<td>Log data manager collect statements</td>
</tr>
<tr>
<td>DRLIxxxx</td>
<td>Component definitions (SQL statements that are executed when the system tables are created or updated)</td>
</tr>
<tr>
<td>DRLLxxxx</td>
<td>Log definitions</td>
</tr>
<tr>
<td>DRLRxxxx</td>
<td>Record definitions</td>
</tr>
<tr>
<td>DRLSxxxx</td>
<td>Table space definitions</td>
</tr>
<tr>
<td>DRLTxxxx</td>
<td>Table and update definitions</td>
</tr>
<tr>
<td>DRLUxxxx</td>
<td>Update definitions (when separate from tables)</td>
</tr>
<tr>
<td>DRLVxxxx</td>
<td>View definitions</td>
</tr>
<tr>
<td>DRLWxxxx</td>
<td>Migration definitions</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Naming convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLOxxxx</td>
<td>Report definitions</td>
</tr>
<tr>
<td>DRLQxxxx</td>
<td>SQL queries</td>
</tr>
<tr>
<td>DRLFxxxx</td>
<td>QMF forms</td>
</tr>
</tbody>
</table>
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.
Collecting log data

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;
//DRLLOG    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:

<table>
<thead>
<tr>
<th>DD statement name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLIN</td>
<td>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.</td>
</tr>
<tr>
<td>DRLLOG</td>
<td>Identifies the log data set. The data set attributes are determined by the program creating the log.</td>
</tr>
<tr>
<td>DRLOUT</td>
<td>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.</td>
</tr>
<tr>
<td>DRLDUMP</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 (DRLLOG 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          *
//* //DRLLOG statement as follows:                                   *
//* //DRLLOG DD DSN=SYSPLEX.OPERLOG,DISP=SHR,                         *
//* DBC=(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 WMPERFT;
/* 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
/* - Insert the correct collect statement in DRLIN (as described above).
/* - Specify the name of the log data set in DRLOG.
/* *********************************************************************************/
/*COLLECT EXEC PGM=DRLPLC,PARM=('SYSTEM=DSN SYSPREFIX=DRLSYS')
/*STEPLIB DD DISP=SHR,DSN=DRL181.SDRLLOAD
/*DRLIN DD DISP=SHR,DSN=DSN710.DSNLOAD
/*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)*/```
Collecting log data

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

<table>
<thead>
<tr>
<th>Collect job name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLJCOIM</td>
<td>Collects IMS log data.</td>
</tr>
<tr>
<td>DRLJCOIN</td>
<td>Collects Tivoli Information Management for z/OS data.</td>
</tr>
<tr>
<td>DRLJCOVP</td>
<td>Collects network configuration data.</td>
</tr>
</tbody>
</table>

**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.

```plaintext
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
DRL0319I ------------------- | ----------
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
DRL0319I Unrecognized | 3
DRL0321I Total | 4582
DRL0003I
DRL0316I Records built by record procedures:
DRL0317I Record name | Number
DRL0319I ------------------- | ----------
DRL0319I SMF_030_X | 2012
DRL0319I SMF_070_X | 200
DRL0321I Total | 2212
DRL0003I
DRL0323I
DRL0324I Table name | Insert Updates | Database Inserts Updates
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 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): 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:
   DRL0317I Record name | Number
   DRL0318I ----------------- | ----------
   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
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:

<table>
<thead>
<tr>
<th>DRL0316I</th>
<th>Records built by record procedures:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL0317I</td>
<td>Record name</td>
</tr>
<tr>
<td>DRL0318I</td>
<td>SMF_030_X</td>
</tr>
<tr>
<td>DRL0319I</td>
<td>SMF_076_X</td>
</tr>
<tr>
<td>DRL0319I</td>
<td>Total</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>DRL0323I</th>
<th>-------Buffer------</th>
<th>------Database------</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL0324I</td>
<td>Table name</td>
<td>Inserts</td>
</tr>
<tr>
<td>DRL0325I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRL0326I</td>
<td>DRL .AVAILABILITY_D</td>
<td>3</td>
</tr>
<tr>
<td>DRL0326I</td>
<td>DRL .MVS_WORKLOAD_M</td>
<td>60</td>
</tr>
<tr>
<td>DRL0327I</td>
<td>Total</td>
<td>2643</td>
</tr>
</tbody>
</table>

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.

![Collect Statistics window](image)

**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:

```plaintext
//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=*  
```

Chapter 6. Setting up operating routines  93
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 nnn 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:
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.

![Diagram of DB2 environment for the Tivoli Decision Support for z/OS database](image)

**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...
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.

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.
DRLJTBSR job that reports tablespace requirements

```plaintext
//DRLJTBSR JOB (ACTC#) , 'SPACE' 00000100
//******************************************************************************
//SYSEXEC 00000200
//********************************************************************
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
//*
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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value to set</th>
<th>Explanation</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBRARY</td>
<td>Tivoli Decision Support for z/OS definition library (UPPERCASE)</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2SUBSYS</td>
<td>DB2 subsystems name (UPPERCASE)</td>
<td>The DB2 subsystem where the product resides. This is a required parameter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSPREFIX</td>
<td>Prefix for system tables (UPPERCASE)</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6. Parameters for tablespace reporting  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value to set</th>
<th>Explanation</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONENT</td>
<td>Component name (UPPERCASE)</td>
<td>The name of a Tivoli Decision Support for z/OS component. This is a required parameter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLENAME</td>
<td>The name of the table (UPPERCASE)</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECNUNBER</td>
<td>Number of rows</td>
<td>The estimated number of rows. This is a required parameter and must be numeric.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAGESIZE</td>
<td>DB2 page size</td>
<td>The DB2 page size. This is an optional parameter; when specified, it must be either 4K or 32K.</td>
<td>4096 (4K)</td>
<td></td>
</tr>
<tr>
<td>MAXROWS</td>
<td>Maximum number of rows per page</td>
<td>The maximum number of rows per page. This is an optional parameter; when specified, it must be a numeric value between 1 and 255.</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>PCTFREE</td>
<td>Percentage of free space on each page</td>
<td>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.</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>FREEPAGE</td>
<td>Number of free space pages</td>
<td>The number of free space pages. This is an optional DB2 parameter; when specified, it must be a numeric value between 1 and 255.</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>COMPRESS</td>
<td>Compression ratio</td>
<td>The compression ratio calculated as PERCSAVE/100 (PERCSAVE is the percentage of kilobytes saved by compression as reported by DB2 utility DSNICOMP). This parameter is optional; when specified, it must be a numeric value.</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

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 DRLSJTBRSR 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
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](imageurl)
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.

Tablespace Other
Select a table space 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.
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:

```sql
//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 Disk for 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.
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.

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

```plaintext
DRL0404I Table name | Deletes
DRL0405I --------------------------- |---------
```

Figure 34. DRLJPURG job that uses all purge conditions
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.

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

```//DRLJCOPY JOB (ACCT#), 'IMAGE COPY' 00010001
//********************************************************************** 00020000
// * LICENSED MATERIALS - PROPERTY OF IBM * 00020002
// * 5698-B06 Copyright IBM Corporation 1992, 2015 * 00040002
// * SEE COPYRIGHT INSTRUCTIONS. * 00050000
// * 00060002
//********************************************************************** 00070002
// * * 00090000
// * NAME: DRLJCOPY 00110002
// * STATUS: Tivoli Decision Support for zOS 1.8.2 00130002
// * 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
// * NOTES: * 00210000
// * CHECK THE FOLLOWING: * 00220002
// * 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: * 00320000
// * CHANGE FLAG TYPE DATE DESCRIPTION * 00340000
// * $D0=DCR066, TDS182,01/06/15,ADL(SM): Update TDS Version and * 00350000
// * DB2 dataset names. * 00360000
// * * 00380000
//********************************************************************** 00400002
//UTIL EXEC DSNUPROC,LIB='db2loadlibrary', 00420002
// SYSTEM=DSN,UID='TEMP',UTPROC='' 00430002
//COPY01 DD DSN=COPYDSN1, 00450002
// DISP=(MOD,CATLG,CATLG), 00460002
// SPACE=(16384,(50,50),,,ROUND), 00470002
// UNIT=SYSDA 00480002
//COPY02 DD DSN=COPYDSN2, 00490002
// DISP=(MOD,CATLG,CATLG), 00500000
// SPACE=(16384,(50,50),,,ROUND), 00510000
// UNIT=SYSDA 00520000
//SYSIN DD * 00530000
COPY TABLESPACE DRLDB.DRLSSYS1 00550000
COPYDN COPY01 00560000
FULL YES 00570000
COPY TABLESPACE DRLDB.DRLSSYS2 00580000
COPYDN COPY02 00590000
FULL YES 00600000
/*
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 table spaces. Ensure that all table spaces 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.

- 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.
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.

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.
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.
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.
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.

**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.

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:

```sql
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:

```sql
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
//********************************************************************
//DRLJBATR
//********************************************************************
// Name: DRLJBATR
00100000
// Status: Tivoli Decision Support for z/OS 1.8.1
00130000
// Function: This job is used to print and/or save all (or a selected
00140000
// subset of) the batch reports.
00190000
// Reports printed to: DSQPRINT with QMF (tables)
00210000
// DRLPRINT w/o QMF (tables) PNA4805
00220000
// Reports saved in : DRLREP (tables)
00240000
// ADMGDF (charts)
00250000
// Messages written to: DRLOUT
00260000
// The exec DRLEBATR accepts the following parameters:
00280000
// SYSTEM=DB2_system DB2 subsystem name. The default is DSN
00300000
// SYSPREFIX=sysprefix Prefix for TDSzOS system
00310000
// tables. The default is DRLSYS.
00320000
// PREFIX=prefix Prefix for all other tables. The default
00330000
// is DRL.
00340000
// SHOWSQL=YES/NO Show SQL statements (for debugging).
00350000
// YES or NO. The default is NO.
00360000
// CYCLE=run_cycle Run cycle: DAILY, WEEKLY or MONTHLY.
00370000
// If not specified, all reports are printed.
00380000
// GROUP=report_group Report group. If not specified, all
00390000
// reports are printed.
00400000
// REPORT=rpt1,rpt2.. Lists the reports to print. If not speci-
00410000
// fied, all reports are printed.
00420000
// PRINTER=prt_name Printer to be used for graphic reports.
00430000
// The default printer is defined in the QMF profile.
00440000
// DIALLANG=n Define the application language. PNA46029
00450000
// 0=1 for English (default) 0=470000
// 2=2 for German 0=480000
// 3=3 for Japanese 0=490000
// QMF=YES/NO Report generation with or w/o QMF. YES or NO.
00500000
// Default is YES.
00510000
// GDDM=YES/NO GDDM available for graphic reports.
00520000
// Default is YES.
00530000
// DRLMAX=nnnn Max number of result rows from a query w/o QMF.
00540000
// Default is 5000.
00550000
// PAGELEN=nn Page length used when printing tabular reports w/o QMF.
00560000
// Default is 60.
00570000
// PAGE=PAGE This word is used in the report footing for page numbering
00570000
// or reports w/o QMF. Default is PAGE.
00570000
// TOTAL=TOTAL This word is used for an across summary column header
00580000
// in tabular reports w/o QMF. Default is TOTAL.
00590000
// DECSEP=PERIOD PERIOD/COMMA. Decimal separator for reports w/o QMF.
00600000
// Default is PERIOD.
00600000
// DUALSAVE=xxx Allow graphic reports to be saved as tabular reports
00600000
// simultaneously.
00600000

114 IBM Tivoli Decision Support for z/OS: Administration Guide and Reference
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 */
/* PRODNAME=Tivoli Decision Supp PN46029 00640002 */
/* This text is used in the report footing. */ 00650002
/* The default is TDSzos */ 00660002
/* Note: If specified, PRODNAME 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 TDSzos data sets. */ 00720002
/* */
/* 2. Change the parameters to DRLEBATR as required. */ 00730002
/* */
/* 3. Remove QMF DD-statements if you are not using QMF. PN48405 */ 00740002
/* Search on 'DSQ' to find such occurrences. PN48405 */ 00750002
/* */
/* The exception is DSQCFRM, which should be changed PN68060 */ 00760000
/* to DRLFORM. The dataset name should point to the */ 00760100
/* user defined forms library. */ 00760200
/* */
/* 4. Change the DB2 load library name according to */ 00780000
/* the naming convention of your installation. */ 00790000
/* Default is 'db2loadlibrary'. */ 00790100
/* */
/* */
/* CHANGE ACTIVITY: */ 00770002
/* */
/* 00 1993-05-18 JHS Created */ 00780002
/* */
/* 01 1993-10-25 JCS Variables DIALLANG and PRODNAME PN46029 */ 00790002
/* */
/* 02 1993-12-01 LW Reporting without QMF PN48405 */ 00800002
/* */
/* 03 1994-11-15 IW DRLFORM DD card added for QMF form */ 00800103
/* and SYSEXEC set to same as SYSPROC PN65801 */ 00810015
/* */
/* 04 1994-12-12 LW Allow 'dual' save PN65801 */ 00810115
/* */
/* 05 1994-12-15 LW ADMPRINTQ added PN65906 */ 00810215
/* */
/* 06 1995-02-16 PN Comment concerning DRLFORM for */ 00810216
/* non-QMF users added above. */ 00810217
/* */
/* 07 2004-11-03 RV Sysroute of apar PQ92756 PQ96265 */ 00811013
/* */
/* */
/* CHANGE ACTIVITY: */
/* */
/* CHANGE FLAG DATE DESCRIPTION */
/* */
/* ***********************************************************/
/* $0=DCR066, TDS180,01/06/07,ADL(SM): Update TDS Version and */
/* $1=DCR116, TDS181,15/05/09,ADL(RC): Update TDS Version */
/* */
/* *********************************************************************************/
/* REPORT EXEC PGM=IKJEFT01 00820000 */
/* */
/* STEPLIB DD DISP=SHR,DSN=DRLvrm,SDRLLOAD */ 00820008
/* */
/* DD DISP=SHR,DSN=qmfloadlibrary */ 00860000
/* */
/* DD DISP=SHR,DSN=db2loadlibrary */ 00870000
/* */
/* SYSPROC DD DISP=SHR,DSN=DRLvrm,SDRELEXEC */ 00880000
/* */
/* DD DISP=SHR,DSN=qmFclstlibrary */ 00890000
/* */
/* SYSEXEC DD DISP=SHR,DSN=DRLvrm,SDRELEXEC */ 00910013
/* */
/* DD DISP=SHR,DSN=qmexeclibrary */ 00900013
/* */
/* *********************************************************************************/
/* MESSAGES */ 00910000
/* */
/* */
/* DRLOUT DD SYSOUT=*/ 00920002
/* *********************************************************************************/
/* PRINT REPORTS TO EITHER DSQPRINT OR DRLPRINT */ 00950000
/* */
/* */
/* DSQPRINT DD SYSOUT=,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330) */ 00960002
/* */
/* */
/* DRLPRINT DD SYSOUT=,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330) */ 00970000
/* */
/* */
/* Chapter 6. Setting up operating routines 115 */
Administering reports

```c
//********************
01000000
//*
01010002
//*
01020000
//DRLREP DD DISP=SHR,DSN=DRL.LOCAL.REPORTS
01030000
//ADMGDF DD DISP=SHR,DSN=DRL.LOCAL.CHARTS
01040000
//*********************
01050000
//*
01060002
//*
01070000
//ADMGGMAP DD DISP=SHR,DSN=ADMGGMAPlibrary
01080000
//ADMCFORM DD DISP=SHR,DSN=ADMCFORMlibrary
01090000
// DD DISP=SHR,DSN=DRLvrm.SDRLFENU
01100008
//ADMSYMBL DD DISP=SHR,DSN=SYS1.GDDMSYM
01110000
//ADMDEFs DD DISP=SHR,DSN=SYS1.GDDMNIcK
01120000
//ADMPRNTQ DD DISP=SHR,DSN=ADMPRINT.REQUEST.QUEUE
01121015
//DSQUCFRM DD DISP=SHR,DSN=DRLvrm.SDRLFENU
01130008
//********************
01140000
//*
01150002
//*
01160000
//DSQDEBUG DD DUMMY
01170000
//DSQUMP DD DUMMY
01180000
//DSQPNL DD DISP=SHR,DSN=QMFDSPQNLxlibrary
01190000
//DSQSPILL DD DSN=&&SPILL,DISP=(NEW,DELETE),UNIT=SYSDA,
01200000
// SPACE=(CYL,(1,1),RLSE),DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096)
01210000
//DSQEDIT DD DSN=&&EDIT,UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
01220000
// DCB=(RECFM=FBA,LRECL=79,BLKSIZE=4029)
01230000
//DRLFORM DD DSN=&&FORMDS,UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
01230312
//DCB=(RECFM=VB,LRECL=255,BLKSIZE=2600),DISP=(NEW,DELETE)
01231010
//********************
01240000
//*
01250002
//*
01260000
//SYSPRINT DD SYSOUT=
01270000
//SYSTSPRT DD SYSOUT=
01280000
//SYSTISIN DD SYSOUT=
01290000
%DRLEBATR SYSTEM=DSN SYSPREFIX=DSYS PREFIX=DRL -
01300000
 PRINTER=XXX -
01310000
 REPORT=XXXXXXX,YYYYYYYY -
01320000
 &SYSTEM_ID='SYS1' -
01330000
 &FROM_DATE='1993-01-01' -
01340000
 &TO_DATE='1993-04-01' -
01350002
 DIALLANG=1
01360003
 PRODNAME=Tivoli Decision Support
01370003
/*
*********************
// QMF LIBRARIES
//*
//ADMGGMAP DD DISP=SHR,DSN=ADMGGMAPlibrary
//ADMCFORM DD DISP=SHR,DSN=ADMCFORMlibrary
// DD DISP=SHR,DSN=DRLvrm.SDRLFENU
//ADMSYMBL DD DISP=SHR,DSN=SYS1.GDDMSYM
//ADMDEFs DD DISP=SHR,DSN=SYS1.GDDMNIcK
//ADMPRNTQ DD DISP=SHR,DSN=ADMPRINT.REQUEST.QUEUE
//DSQUCFRM DD DISP=SHR,DSN=DRLvrm.SDRLFENU
//*********************
// START EXEC DRLEBATR
//*
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>
<thead>
<tr>
<th>Parameter</th>
<th>Value to set</th>
<th>Explanation</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>DB2 subsystem name (UPPERCASE)</td>
<td>The DB2 subsystem where Tivoli Decision Support for z/OS resides.</td>
<td>DSN</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7. Parameters for batch reporting (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value to set</th>
<th>Explanation</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPREFIX</td>
<td>Prefix for system tables (UPPERCASE)</td>
<td>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.</td>
<td>DRLSYS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This required parameter can be 8 alphanumeric characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is DRLSYS. If the value is something other than DRLSYS, it was changed during installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not change the value; Tivoli Decision Support for z/OS uses this value to access its system tables.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREFIX</td>
<td>Prefix for all other tables (UPPERCASE)</td>
<td>The prefix of Tivoli Decision Support for z/OS data tables in the DB2 database.</td>
<td>DRL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid values are determined at installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This required parameter can be 8 alphanumeric characters. The first character must be alphabetic.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is DRL. If the value is something other than DRL, it was changed during installation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHOWSQL</td>
<td>YES or NO (UPPERCASE)</td>
<td>Here you specify if SQL statements should be shown (for debugging purposes).</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>CYCLE</td>
<td>DAILY, WEEKLY or MONTHLY (UPPERCASE)</td>
<td>The run cycle for reports. If you do not specify daily, weekly, or monthly, all reports are printed.</td>
<td>All reports</td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>A report group ID (UPPERCASE)</td>
<td>Here you can specify the ID of a report group. If you do not specify a group, all reports are printed.</td>
<td>All reports</td>
<td></td>
</tr>
<tr>
<td>REPORT</td>
<td>One or more report IDs (UPPERCASE)</td>
<td>Here you can specify one or more reports to be printed. If you do not specify any reports, all reports are printed.</td>
<td>All reports</td>
<td></td>
</tr>
<tr>
<td>PRINTER</td>
<td>Default printer name (UPPERCASE)</td>
<td>The GDDM nickname of a printer to use for printing graphic reports. The printer should be capable of printing GDDM-based graphics.</td>
<td>As defined in the QMF profile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The printer name must be defined in the GDDM nicknames file, allocated to the ADMDEFS ddbname. Refer to the QMF: Reference and GDDM User’s Guide for more information about defining GDDM nicknames.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter cannot be used if QMF=NO.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIALLANG</td>
<td>1. English 2. Japanese</td>
<td>With this parameter, you specify the language to be used.</td>
<td>1=English</td>
<td></td>
</tr>
<tr>
<td>QMF</td>
<td>YES or NO (UPPERCASE)</td>
<td>With this parameter, you specify whether your installation uses QMF or not.</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>GDDM</td>
<td>YES or NO (UPPERCASE)</td>
<td>With this parameter, you specify if your installation uses GDDM.</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>DRLMAX</td>
<td>nnnn</td>
<td>If your installation does not use QMF, you use this parameter to specify the maximum number of result rows from a query.</td>
<td>5000</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Parameters for batch reporting (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value to set</th>
<th>Explanation</th>
<th>Default value</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGELEN</td>
<td>mm</td>
<td>If your installation does not use QMF, you use this parameter to specify the page length when printing tabular reports.</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>PAGE</td>
<td>The word for page (Mixed case)</td>
<td>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.</td>
<td>PAGE</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>The word for total (Mixed case)</td>
<td>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.</td>
<td>TOTAL</td>
<td></td>
</tr>
<tr>
<td>DECSEP</td>
<td>Period or comma</td>
<td>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.</td>
<td>PERIOD</td>
<td></td>
</tr>
<tr>
<td>DUALSAVE</td>
<td>YES or NO (UPPERCASE)</td>
<td>Allow graphic reports to be saved as tabular reports simultaneously.</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>&amp;variable</td>
<td>A value</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRODNAME</td>
<td>Tivoli Decision Support for z/OS Report (Mixed case)</td>
<td>This text is used in the report footer. If specified, PRODNAME must be the last parameter.</td>
<td>Tivoli Decision Support for z/OS Report</td>
<td></td>
</tr>
</tbody>
</table>

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.
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.
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):

```
:hl. 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.
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
******************************************************************************
ﾉ='/ LICENSED MATERIALS - PROPERTY OF IBM 00020000
ﾉ'/ = 5698-B06 COPYRIGHT IBM CORPORATION 1993, 2015 00030000
ﾉ'/ = SEE COPYRIGHT INSTRUCTIONS. 00040004
ﾉ'/ = 00050000
ﾉ'/ = 00060004
ﾉ'/ = 00070004
ﾉ'/ = 00080000
******************************************************************************
ﾉ'/ = 00090000
ﾉ'/ = 00100000
ﾉ'/ = 00110004
ﾉ'/ = 00120000
ﾉ'/ = 00130004
ﾉ'/ = 00140000
ﾉ'/ = 00150004
ﾉ'/ = 00160004
ﾉ'/ = 00170004
ﾉ'/ = 00180004
ﾉ'/ = 00190004
ﾉ'/ = 00200004
ﾉ'/ = 00210004
ﾉ'/ = 00220004
ﾉ'/ = 00230000
ﾉ'/ = 00240004
ﾉ'/ = 00250004
ﾉ'/ = 00260004
ﾉ'/ = 00270004
ﾉ'/ = 00280004
ﾉ'/ = 00290004
ﾉ'/ = 00300004
ﾉ'/ = 00310000
ﾉ'/ = 00320004
ﾉ'/ = 00330004
ﾉ'/ = 00340004
ﾉ'/ = 00350004
ﾉ'/ = 00360000
ﾉ'/ = 00370004
ﾉ'/ = 00380004
ﾉ'/ = 00390004
ﾉ'/ = 00400004
ﾉ'/ = 00410004
ﾉ'/ = 00420000
ﾉ'/ = 00430000
ﾉ'/ = 00440000
ﾉ'/ = 00450000
ﾉ'/ = 00460000
ﾉ'/ = 00470002
ﾉ'/ = 00480000
ﾉ'/ = 00490002
ﾉ'/ = 00500000

DRLJEXCE job for generating problem records

---

IBM Tivoli Decision Support for z/OS: Administration Guide and Reference

122
Administering problem records

//* TIVOLI SERVICE DESK LIBRARIES
//************************************************************************
//BLGSD  DD DISP=SHR,DSN=TSD.SDDS
//BLGSI  DD DISP=SHR,DSN=TSD.SDIOS
//BLGSL  DD DISP=SHR,DSN=TSD.SDLDS
//BLGPNL0 DD DISP=SHR,DSN=TSD.IBMPNLS
//BLGPNL1 DD DISP=SHR,DSN=TSD.RPANEL1
//BLMFMT DD DISP=SHR,DSN=TSD.BLMFMT
//ISPLLIB DD DISP=SHR,DSN=TSD.SBLMMOD1
//************************************************************************
//DRLOUT DD SYSOUT=* 00630000
//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:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Book name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 Performance</td>
<td>AS/400 System Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>CICS Performance</td>
<td>CICS Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>Distributed Systems Performance</td>
<td>Distributed Systems Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>IMS Performance</td>
<td>IMS Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>Network Performance</td>
<td>Network Performance Feature Installation and Administration</td>
</tr>
<tr>
<td>System Performance</td>
<td>System Performance Feature Guide</td>
</tr>
<tr>
<td>Resource Accounting</td>
<td>Resource Accounting for z/OS</td>
</tr>
</tbody>
</table>

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.
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

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
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**

<table>
<thead>
<tr>
<th>Line</th>
<th>Log Collector Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>93</td>
<td>DRL0125I The record SMF_080 is defined.</td>
</tr>
<tr>
<td>96</td>
<td>DRL0130I The comment is stored for the record SMF_080.</td>
</tr>
<tr>
<td>1007</td>
<td>DRL0201I The update RACF_COMMAND_80 is defined.</td>
</tr>
<tr>
<td>1014</td>
<td>DRL0403I The purge condition for DRL .RACF_COMMAND_T is added.</td>
</tr>
<tr>
<td>1138</td>
<td>DRL0201I The update RACFLOGON_80 is defined.</td>
</tr>
<tr>
<td>1145</td>
<td>DRL0403I The purge condition for DRL .RACF_LOGON_T is added.</td>
</tr>
<tr>
<td>1293</td>
<td>DRL0201I The update RACFOPERATION_80 is defined.</td>
</tr>
<tr>
<td>1300</td>
<td>DRL0403I The purge condition for DRL .RACF_OPERATION_T is added.</td>
</tr>
<tr>
<td>1466</td>
<td>DRL0201I The update RACFRESOURCE_80 is defined.</td>
</tr>
<tr>
<td>1473</td>
<td>DRL0403I The purge condition for DRL .RACF_RESOURCE_T is added.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line</th>
<th>Report Definition Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1503</td>
<td>DRL3001I The group RACF is defined.</td>
</tr>
<tr>
<td>1511</td>
<td>DRL3001I The report RACF01 is defined.</td>
</tr>
<tr>
<td>1519</td>
<td>DRL3001I The report RACF02 is defined.</td>
</tr>
<tr>
<td>1527</td>
<td>DRL3001I The report RACF03 is defined.</td>
</tr>
<tr>
<td>1535</td>
<td>DRL3001I The report RACF04 is defined.</td>
</tr>
<tr>
<td>1543</td>
<td>DRL3001I The report RACF05 is defined.</td>
</tr>
<tr>
<td>1551</td>
<td>DRL3001I The report RACF06 is defined.</td>
</tr>
<tr>
<td>1559</td>
<td>DRL3001I The report RACF07 is defined.</td>
</tr>
</tbody>
</table>

---

**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].
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.

3. 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.

4. 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.

5. Edit any other lookup tables that the component requires.

When you finish, the installation is complete.

6. 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.

7. 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.

**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.
   
   The Select Table window is displayed (Figure 47 on page 132).
Installing and uninstalling a component

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=Blkwd
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.

<table>
<thead>
<tr>
<th>Message</th>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLCODE=-204 name IS AN UNDEFINED NAME</td>
<td>RC=8</td>
<td>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.</td>
</tr>
<tr>
<td>SQLCODE=+562 A GRANT OF A PRIVILEGE WAS IGNORED BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR</td>
<td>RC=4</td>
<td>You can always ignore this message and return code.</td>
</tr>
<tr>
<td>SQLCODE=-601 THE NAME OF THE OBJECT TO BE CREATED IS IDENTICAL TO THE EXISTING NAME name OF THE OBJECT TYPE objecttype</td>
<td>RC=8</td>
<td>You can always ignore this message and return code.</td>
</tr>
<tr>
<td>SQLCODE=-612 column name IS A DUPLICATE COLUMN NAME</td>
<td>RC=8</td>
<td>You can always ignore this message and return code.</td>
</tr>
</tbody>
</table>

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:
   
<table>
<thead>
<tr>
<th>Feature name</th>
<th>Book name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 Performance</td>
<td>AS/400 System Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>CICS Performance</td>
<td>CICS Performance Feature Guide and Reference</td>
</tr>
<tr>
<td>Distributed Systems Performance</td>
<td>Distributed Systems Performance Feature Guide and Reference, SH19-4018</td>
</tr>
<tr>
<td>IMS Performance</td>
<td>IMS Performance Feature Guide and Reference, SH19-6825</td>
</tr>
<tr>
<td>Network Performance</td>
<td>Network Performance Feature Reference</td>
</tr>
<tr>
<td>System Performance</td>
<td>System Performance Feature Reference</td>
</tr>
<tr>
<td>For Resource Accounting, see</td>
<td>Resource Accounting for z/OS, SH19-4495 book.</td>
</tr>
</tbody>
</table>

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 SDRLDEFS 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.

   ![Figure 49. Component window](image)

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
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)
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.
2. 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].

![Figure 50. Data Sets window](image)

![Figure 51. Collect Statistics window](image)

3. 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.
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).
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).

      The Logs window is displayed.

Figure 53. Sample log statistics output
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.

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.

---

**Figure 54. Record Data window**

Press Enter to view the next record.

Record name . : SMF_030 Record number : 3
Data set . . : LDG.SMFSYSA.W20

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Length</th>
<th>Offset</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF30LEN</td>
<td>BINARY</td>
<td>2</td>
<td>0</td>
<td>628</td>
</tr>
<tr>
<td>SMF30SEG</td>
<td>BINARY</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>SMF30FLG</td>
<td>BIT</td>
<td>1</td>
<td>4</td>
<td>11001110</td>
</tr>
<tr>
<td>SMF30RTY</td>
<td>BINARY</td>
<td>1</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>SMF30TME</td>
<td>TIME</td>
<td>4</td>
<td>6</td>
<td>07.00.03.830000</td>
</tr>
<tr>
<td>SMF30DTE</td>
<td>DATE</td>
<td>4</td>
<td>10</td>
<td>2000-06-03</td>
</tr>
<tr>
<td>SMF30SID</td>
<td>CHAR</td>
<td>4</td>
<td>14</td>
<td>MVS1</td>
</tr>
<tr>
<td>SMF30WID</td>
<td>CHAR</td>
<td>4</td>
<td>18</td>
<td>JES2</td>
</tr>
<tr>
<td>SMF30STP</td>
<td>BINARY</td>
<td>2</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>TRIPLETS</td>
<td>SECTION</td>
<td>.....</td>
<td>.....</td>
<td>(1)</td>
</tr>
<tr>
<td>SMF30SOF</td>
<td>BINARY</td>
<td>4</td>
<td>0</td>
<td>112</td>
</tr>
</tbody>
</table>

---

Command ==>

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

---

Chapter 8. Working with definitions 149
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](image)

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.

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 Definition window](image)

   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).
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).
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.
Working with record definitions in a log

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).
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).

   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.
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](#)).
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

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.

```
REVIEW  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     001    Sys Supp  PIANKA     40       267
2000-01-01  15.00.00  SYS1     002    Appl Dev  ADAMS     72       198
2000-01-02  08.00.00  SYS1     003    Appl Dev  JONES     28       131
2000-01-02  11.00.00  SYS1     004    Retail    PEREZ      21       171
2000-01-03  10.00.00  SYS1     005    Marketing KANG     74       220
2000-01-03  11.00.00  SYS1     006    Manufact  LEE       22       234
2000-01-03  11.00.00  SYS1     007    Manufact  LUTZ      2        95
2000-01-04  07.00.00  SYS1     008    Finance   HAAS      26       109
2000-01-04  07.00.00  SYS1     009    Sys Supp  THOMPSON  84        64
2000-01-04  08.00.00  SYS1     010    Marketing KANG     63       290
2000-01-04  08.00.00  SYS1     011    Finance   GEYER     94       131
2000-01-04  08.00.00  SYS1     012    Finance   GEYER     94       131
2000-01-04  09.00.00  SYS1     013    Marketing STERN    51       162
2000-01-04  09.00.00  SYS1     014    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.  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.
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.
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 Size window](image)

3. After you finish viewing this window, press Enter.
   You are returned to the Tables 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.

F1=Help F2=Split F7=Bkwd F8=Fwd F9=Swap F12=Cancel
Command ===>
Table Maintenance Utilities Edit View Other Help
Press Enter to return.

Table Size

<table>
<thead>
<tr>
<th>Name</th>
<th>Prefix</th>
<th>Row</th>
<th>Row length</th>
<th>Kbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVSPM_DEVICE_H</td>
<td>DRL</td>
<td>80927</td>
<td>240</td>
<td>18967</td>
</tr>
<tr>
<td>MVSPM_DEVICE_AP_H</td>
<td>DRL</td>
<td>34821</td>
<td>102</td>
<td>34688</td>
</tr>
<tr>
<td>MVSPM_CHANNEL_H</td>
<td>DRL</td>
<td>9338</td>
<td>140</td>
<td>1276</td>
</tr>
<tr>
<td>MVSPM_APPL_H</td>
<td>DRL</td>
<td>2388</td>
<td>491</td>
<td>1145</td>
</tr>
<tr>
<td>MVSPM_WORKLOAD_H</td>
<td>DRL</td>
<td>2727</td>
<td>308</td>
<td>820</td>
</tr>
<tr>
<td>MVSPM_STORAGE_H</td>
<td>DRL</td>
<td>2567</td>
<td>199</td>
<td>498</td>
</tr>
<tr>
<td>MVSPM_PAGE_DS_H</td>
<td>DRL</td>
<td>966</td>
<td>229</td>
<td>216</td>
</tr>
<tr>
<td>MVSPM_XCF_PATH_H</td>
<td>DRL</td>
<td>1296</td>
<td>171</td>
<td>216</td>
</tr>
<tr>
<td>MVSPM_SWAP_H</td>
<td>DRL</td>
<td>1771</td>
<td>114</td>
<td>197</td>
</tr>
<tr>
<td>MVSPM_ENQUEUE_H</td>
<td>DRL</td>
<td>1642</td>
<td>100</td>
<td>160</td>
</tr>
</tbody>
</table>

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).

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.
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](image)

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).
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.

---

**Figure 69. Column Values window**

The table shows the column values window with the Recalculate function enabled and the Condition row highlighted. Columns include:

- TIME
- SYSTEM_ID
- DEPARTMENT_NAME
- USER_ID
- TRANSACTIONS
- RESPONSE_SECONDS
- CPU_SECONDS

The values for each column are:

- TIME: 
- SYSTEM_ID: 
- DEPARTMENT_NAME: 
- USER_ID: 
- TRANSACTIONS: 
- RESPONSE_SECONDS: 2.0
- CPU_SECONDS: 2.0

The command line at the bottom of the window shows the available keyboard shortcuts for navigating and editing within the 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](image)

   2. Select option Unload, from the Utilities pull-down menu.

   The Unload Utility window opens, as shown in the following figure:
3. From the Unload Utility window, specify the unload type by inserting 1 for disk unload or 2 for tape unload. The default is DiskUnload.

4. Specify the name of the table and data set you want to unload.

5. If you selected Disk Unload:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the data set already exists</td>
<td>Leave the fields UNIT and VOLSER blank. If you need to create a new data set, enter the required information in both the fields.</td>
</tr>
<tr>
<td>If you selected Tape Unload</td>
<td>Specify the tape unit in the UNIT field, and the tape label in the VOLSER field.</td>
</tr>
</tbody>
</table>

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
// UTILTY.
// THEREFORE, PLEASE, NOTE THAT THIS IS ONLY
// A SAMPLE THAT NEEDS TO BE PROPERLY CUSTOMIZED.
// WARNINGS:
// PLEASE CHECK PLAN NAME (NORMALLY DSNTIBVR),
// 'SQL' 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.
//
// IMPORTANT:
// CHECK THE DATA SET PARAMETER IF YOU HAVE CHosen
// THE UNLOAD ON TAPE.
//
//UNLOAD  EXEC PGM=IKJET01,DYNAMNBR=20
//SYSTSPT DD SYSOUT=
//SYSTSIN DD *
DSN SYSTEM(DSN6)
RUN PROGRAM(DSNTIAUL) PLAN(DSNTIB71) -
   PARM('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,
```
The following example shows control statements for the Load utility. Data is loaded from tape into the AVAILABILITY_D table. The DDNAME for the SYSIN 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.
/* DSNTIAUT 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 SYSUP).
/* as USERID.SYSUP (USERID.SYSUP).
/* SYSREC00 DATASET IS SELECTED FROM THE PREVIOUS PANEL
/*
/*
/* IMPORTANT:
/* SYSUP 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.SYSUP,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.
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 (INZUTLIB)
//** 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.
//**
//** IMPORTANT:
//** CHECK THE DATA SET PARAMETER IF YOU HAVE CHosen
//** THE UNLOAD ON TAPE.
//**
```

---

**Figure 72. DB2 High Performance 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..................
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.
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.
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].

Figure 74. Column Definition 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.

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.
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.

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).
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

1. From the Indexes window, select the index and press F11.
   A confirmation window is displayed.
2. 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](image)

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](image)

   **Complete these fields in the window:**

   - **Source** . . : SAMPLE_01
   - **Prefix** : (blank for record)
   - **Target** . . : SAMPLE_H
   - **Prefix** : DRL
   - **Condition** . . : 
   - **Comments** . . :

   **Column** | **Function** | **Expression**
   --- | --- | ---
   DATE | SO1DATE | >
   TIME | ROUND(SO1TIME, 1 HOUR) | >
   SYSTEM_ID | SO1SYS | >
   DEPARTMENT_NAME | VALUE(LOOKUP DEPARTMENT_NAME IN DRL) | >
   USER_ID | SO1USER | >
   TRANSACTIONS | SUM | SO1TRNS | >
   RESPONSE_SECONDS | SUM | SO1RESP | >
   CPU_SECONDS | SUM | SO1CPU/100.0 | >
   PAGES_PRINTED | SUM | SO1PRNT | >

   **Command ===>>**
   F1=Help  F2=Split  F3=Exit  F4=Prompt  F5=Abbrev  F6=Distrib
   F7=8kwd  F8=Fwd  F9=Swap  F10=Show fld  F11=Schedule  F12=Cancel

---

Working with tables and update definitions

Chapter 9. Working with tables and definitions  185
**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).

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:
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).
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).
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.

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.
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).
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.

Tivoli Decision Support for z/OS takes action depending on the parameters to be changed:
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>
<thead>
<tr>
<th>Index Utilities Other Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Open... Enter 2. Delete... 3. Print List... 4. Exit</td>
</tr>
</tbody>
</table>

Indexes in SVTDB database Row 1 to 21 of 443

<table>
<thead>
<tr>
<th>Table name</th>
<th>Indexspace</th>
<th>Unique</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILD_IX</td>
<td>AVAILABILITY_D</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>AVAILM_IX</td>
<td>AVAILABILITY_M</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>AVAILP_IX</td>
<td>AVAILABILITY_PARM</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>AVAILT_IX</td>
<td>AVAILABILITY_T</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>AVAILW_IX</td>
<td>AVAILABILITY_W</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Command ===>
F1=Help F2=Split F3=Exit F7=Bkwd F8=Fwd F9=Swap
F10=Actions 12=Cancel

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 be changed:

Where the index must be recreated
In this case the product:
   a. Asks you to confirm the change.
   b. Deletes the index, with the DROP command.
   c. 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:
   a. Stops the database.
   b. Makes the change, using the ALTER command.
   c. 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:
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.

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 Tablespace 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].
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.
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>
<thead>
<tr>
<th>Table name</th>
<th>Prefix</th>
<th>Database</th>
<th>Tablespace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DRL</td>
<td>DRLDB</td>
<td></td>
</tr>
</tbody>
</table>

Comments

/ Column Type Length Nulls Primary Key

******************************************************************************

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 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.
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](image)

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.
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.

---

### Figure 94. Revoke Privilege window

<table>
<thead>
<tr>
<th>User ID</th>
<th>Table</th>
<th>Privilege</th>
<th>Grantor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL</td>
<td>DRL.SAMPLE_H</td>
<td>DELETE</td>
<td>DRL</td>
</tr>
<tr>
<td>DRL</td>
<td>DRL.SAMPLE_H</td>
<td>UPDATE</td>
<td>DRL</td>
</tr>
<tr>
<td>DRL</td>
<td>DRL.SAMPLE_H</td>
<td>INSERT</td>
<td>DRL</td>
</tr>
<tr>
<td>DRLUSER</td>
<td>DRL.SAMPLE_H</td>
<td>SELECT</td>
<td>DRL</td>
</tr>
<tr>
<td>DRLUSER</td>
<td>DRL.SAMPLE_H</td>
<td>SELECT</td>
<td>DRL</td>
</tr>
</tbody>
</table>

Command ===>

F1=Help  F2=Split  F7=Bkwd  F8=Fwd  F9=Swap  F12=Cancel
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 successfully collected log data sets” on page 216).
- 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
data manager Main Selection window (Figure 95) is displayed.

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 DRLLOG DD statement of the job step (described in 207).
5. Run the job you have now amended, to create the log data set.
Job step for recording a log data set for collection

DRLJLDML sample job

This job is shipped with Tivoli Decision Support for z/OS as sample job DRLJLDML.

DRLJLDML job
/DRRLJLDML JOB (ACCT#), 'LOGS' 00010014
/**************************************************************************
* Licensed Materials - Property of IBM
* * 00300000
* * 00440000
* * 00500000
* * 0698-806 Copyright IBM Corporation 1995, 2015
* * See Copyright instructions.
* * 00600000
* * 00700000
* * 00800000
**************************************************************************
** Name: DRLJLDML */ 00090000
** Status: Tivoli Decision Support for zOS 1.8.2 */ 00130011
** Function: */ 00140000
** Log Data Manager - register a log data set sample job */ 00160014
** This job is used to register the log data set (only one) */ 00170000
** specified in DRLOG in the DRLDM_LOGDATASETS as being ready */ 00180011
** for collect by the Log Data Manager. */ 00190000
** Input: */ 00191000
** The exec DRLJLDML accepts the following parameters: */ 00192000
** SYSPREFIX=xxxxxxx Prefix for system tables. default=DRLSYS */ 00200000
** PLAN=xxxxxxx DB2 plan name default=DRLPLAN */ 00201000
** LOGTYPE=xxxxxxx Log type (e.g. SMF). Required. */ 00202000
** LOGID=xxxxxx Log ID. If not specified (or ='), a blank */ 00203000
** statement is used in collect. */ 00204000
** ONTape=N/Y Specify if the LOG name is on DASD or not. If* */ 00205000
** not coded, it defaults to NO. */ 00206000
** DRLOG DD card: Name of log data set to be registered */ 00207000
** (can refer to a previous step). */ 00208000
** It must be cataloged. */ 00209000
** Output: Log data set name registered in */ 00210000
** sysprefix.DRRLDM_LOGDATASETS together */ 00211000
** with LOG_NAME, LOG_ID and TIME_ADDED. */ 00212000
** Confirmation message including data set name */ 00213000
** Notes: */ 00214000
** Before you submit the job, do the following: */ 00215000
** 1. Fill in a correct log data set name. */ 00216000
** 2. Check that the steplib db2loadlibrary is correct. */ 00217000
** 3. Change the input parameters to DRLJLDML as required. */ 00218000
** 4. Change the DB2 load library name according to */ 00219000
** the naming convention of your installation. */ 00220000
** Default is 'db2loadlibrary'. */ 00221000
** 5. Change the TDSz high level qualifier. Default id 'DRLvrm'. */ 00222000
** CHANGE ACTIVITY: */ 00223000
** 00 1995-03-05 IW Created */ 00224000
** 01 1997-05-14 GL Added OMPAPE parameter to allow PQ06678 */ 00225000
** the use of tape log names PQ06678 */ 00226000
** 07 1997-05-14 GL Added OMPAPE parameter to allow PQ06678 */ 00227000
**
Job step for recording a log data set for collection

```c
/* 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 */
/* ➔ */
/* ➔-----------------------------------------------------------------*/
```  

```c
00780000 //LDMLOG EXEC PGM=IKJEFT01 00790000 //SYSPROC DD DISP=SHR,DSN=DRLvrm.SDRLEXEC 00800011 //STEPLIB DD DISP=SHR,DSN=DRLvrm.SDRLLOAD 00800211 // DD DISP=SHR,DSN=db2loadlibrary <-- 00810010 //********************************************** 00870010 // MESSAGES 00880000 // ➔ 00890000 //DRLOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80) 00900009 //********************************************** 00930010 // LOG DATA SET 00940000 // ➔ DSN=*.stepname.ddname can be used 00950000 // ➔ 00960010 //DRLOG DD DISP=SHR,DSN=... <-- 00970010 //********************************************** 00980010 // START EXEC DRLELDML 01210002 // ➔ 01220000 //SYSPRINT DD SYSOUT=* 01230000 //SYSTSPRT 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 21
- "Changing the collect statements data set name" on page 21

---

**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 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.

---

**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.

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.

![Add Collect Statements Definition window](image)

**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.
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:
If there are many log data sets with status 'SELECT', run job DRLLELDLC 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 DRLLELDLC is run.

---

**Modifying log collector statements**

---

**Modifying log collector statements**

---

**Modifying log collector statements**

---

**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

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 DRIJLDJMC 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.
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 “DRLLDM_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 DRLJLDM (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'
//*************************************************************************
//* Licensed Materials - Property of IBM
//* 5698-B06 Copyright IBM Corporation 2003, 2015
//* See Copyright instructions.
//*************************************************************************
//* Name: DRLJLDMC
//* Status: Tivoli Decision Support for zOS 1.8.2
//* Function:
//* Log Data Manager Collect Log Data Sets sample job
//* This job is used to collect log data sets that are recorded
//* in the DRLDM_LOGDATASETS system table as being ready for
//* collect by the Log Data Manager.
//* Input:
//* The exec DRLELDMC accepts the following parameters:
//* SYSPREFIX=xxxxxxx Prefix for system tables. default=DRLSYS  * 00196000
//* SYSTEM=xxxxxx DB2 subsystem name. default=DSN  * 00197000
//* PREFIX=xxxxxxx Prefix for all other tables.default=DRL  * 00198000
//* PLAN=xxxxxxx DB 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=xxxxxxxxxx Log type (e.g. SMF). If not specified, all log types are selected for processing. * 00199100
//* 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
//* named 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 counted* * 00261000
//* or viewed by the Log Data Manager dialog. * 00270000
//* * 00270100
//* DRLLOG DD card: Must not be allocated. * 00270200
//* ** 00270300
//* LMLOG 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 log data sets to be collected. * 00270500
// Output: The results of the collects are recorded in sysprefix.DRLLOM_LOGDATASETS together with LOG_NAME, LOG_ID and TIME_ADDED. * 00290000
// with LOG_NAME, LOG_ID and TIME_ADDED. * 00291000
// Job messages in the DRLMSG file * 00291200
// * 00291300
// Notes: Before you submit the job, do the following: * 00291400
// 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 the naming convention of your installation. * 00292200
// 4. Change the TDSz data set HLQ (default is DRLvm.) * 00292401
// * 00292500
// CHANGE ACTIVITY: * 00292600
// 00 1995-03-05 I..W Created * 00292700
// * 00293000
// CHANGE ACTIVITY: * 00293201
// CHANGE FLAG TYPE DATE DESCRIPTION * 00293301
// **************************************************************************
// $D0-DOR006, TDSIBO,01/06/07,ADL(SM): Update TDS Version and DB2 dataset names. * 00293500
// $D1-DOR116, TDSIB1,15/05/09,ADL(RC): Update TDS Version * 00293701
// * 00293801
// **************************************************************************
// EXEC PGM=IKJEFT01, DYNAMNBR=20 00294000
// SYSPROC DD DISP=SHR, DSN=DRLvm.SDRELCE -- 00297001
// STEPLIB DD DISP=SHR, DSN=DRLvm.SDRDLLOAD -- 00298000
// DD DISP=SHR, DSN=db2loadlibrary -- 00299000
// **************************************************************************
// DRLOUT DD SYOUT=*, DCB=(RECFM=F, LRECL=80) 00300100
// DRLDUMP DD SYOUT=*, DCB=(RECFM=F, LRECL=80) 00302000
// **************************************************************************
// MESSAGES 00320000
// * 00330000
// DRLMSG DD SYOUT=*, DCB=(RECFM=F, LRECL=80) 00331000
// **************************************************************************
// Add the next three DD statements if you collect IMS. 00331400
// Note 1: IMSVER must specify the same release as the collect statement used by the Log Data Manager. 00331700
// Note 2: DRLICHKI must be DUMMY or point out an empty data set after an IMS restart. 00331900
// **************************************************************************
// 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
// **************************************************************************
// START EXEC DRLELDMC 00333000
// * 00410000
// SYSPRINT DD SYSOUT=* 00440000
// SYSTSPRT DD SYSOUT=* 00450000
// SYSTSN DD * 00460000
// %DRLELDMC SYSTEM=DSN SYSPREFIX=DRLSYS PREFIX=DRL - 00470000
// DSPREFIX=DRL - 00480000
// LOGTYPE=SMF - 00481000
// LOGID='' 00490000
// */
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.

DRLLOG 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.

<table>
<thead>
<tr>
<th>Log data set</th>
<th>Time collected</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS170.SMFLOGX.SLOG950120</td>
<td>2004-11-21.02.01.25</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOGB.SLOG950120</td>
<td>2004-11-21.01.33.25</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOGA.SLOG950120</td>
<td>2004-11-21.01.15.10</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOG.SLOG950120B</td>
<td>2004-11-21.01.01.20</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOG.SLOG950120A</td>
<td>2004-11-21.00.45.20</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOG.SLOG950119</td>
<td>2004-11-20.23.15.10</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOG.SLOG950119B</td>
<td>2004-11-20.01.45.20</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOGB.SLOG950119</td>
<td>2004-11-20.01.13.25</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOG.SLOG950119A</td>
<td>2004-11-20.01.13.25</td>
<td>0</td>
</tr>
<tr>
<td>SYS170.SMFLOG.SLOG950119A</td>
<td>2004-11-20.00.45.20</td>
<td>0</td>
</tr>
</tbody>
</table>

Command ==> ___________________________

F1=Help  F2=Split  F3=Exit  F5=DRLDUMP  F6=Retent.  F7=Back
F8=Forward  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 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.

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.

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT_TYPE</td>
<td>K</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Object type. This is LOG, RECORD, RECPROC, or UPDATE.</td>
</tr>
<tr>
<td>OBJECT_NAME</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td>EXPRESSION_NO</td>
<td>K</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>EXPRESSION</td>
<td></td>
<td>VARCHAR(2000)</td>
</tr>
<tr>
<td>PARSED_EXPRESSION</td>
<td>V</td>
<td>VARCHAR(2000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parsed version of the expression.</td>
</tr>
</tbody>
</table>

**DRLFIELDS**

This system table contains one row for every field in each defined record type.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_NAME</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name of the record. For a log header, this is &quot;log-name&quot;.</td>
</tr>
<tr>
<td>FIELD_NO</td>
<td>K</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>FIELD_NAME</td>
<td></td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Name of the field.</td>
</tr>
</tbody>
</table>
## Log collector system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>CHAR(8)</td>
<td>Type of the field. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Type</strong>  <strong>Field Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BINARY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BINARYS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EINTEGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HEXIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DECIMAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZONED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FLOAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHAR(*)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VARCHAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DATE_007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIME_007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NTV_001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSTAMP_1</td>
</tr>
<tr>
<td>LENGTH</td>
<td>SMALLINT</td>
<td>Length of the field. For DECIMAL and ZONED fields, this is a 1-byte precision followed by a 1-byte scale.</td>
</tr>
<tr>
<td>OFFSET</td>
<td>SMALLINT</td>
<td>Offset of the field in the record or section.</td>
</tr>
<tr>
<td>INSECTION_NO</td>
<td>SMALLINT</td>
<td>Number of the section where the field is contained. This is zero if the field is not in a section.</td>
</tr>
<tr>
<td>REMARKS</td>
<td>VARCHAR(254)</td>
<td>Description of the field, set by the COMMENT ON statement.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_NAME</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td>LOG_ID</td>
<td>K</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>COLLECT_STMT_DS</td>
<td>VARCHAR(54)</td>
<td></td>
</tr>
</tbody>
</table>
## DRLLOGDATASETS

This system table contains one or more rows for each log data set recorded by the Log Data Manager.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATASET_NAME</td>
<td>VARCHAR(54)</td>
<td>Name of the log data set, including the member name (for a PDS member).</td>
</tr>
<tr>
<td>LOG_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the log type.</td>
</tr>
<tr>
<td>TIME_COLLECTED</td>
<td>TIMESTAMP</td>
<td>Timestamp of the collect. For a data set not yet collected it is 0001-01-01-00.00.00.000000. For a successfully collected data set it is set to the value of the TIME_COLLECTED field in the corresponding entry in DRLLOGDATASETS. 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.</td>
</tr>
<tr>
<td>LOG_ID</td>
<td>CHAR(8)</td>
<td>The log ID currently associated with this data set.</td>
</tr>
<tr>
<td>TIME_ADDED</td>
<td>TIMESTAMP</td>
<td>Timestamp when the log data set was first recorded.</td>
</tr>
<tr>
<td>TIME_COLLECT_CALL</td>
<td>TIMESTAMP</td>
<td>Timestamp when the DRLELDMC exec called the log collector to process the log data set.</td>
</tr>
<tr>
<td>COLLECT_RC</td>
<td>CHAR(5)</td>
<td>The return code from the collect. It is blank if not yet collected; '0' or '4' if successfully collected; '&gt;= 8' if unsuccessfully collected without abend; 'Unn' if the collect ended with a user abend; 'Snn' if the collect ended with a system abend.</td>
</tr>
<tr>
<td>OUTPUT_DS</td>
<td>VARCHAR(35)</td>
<td>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.</td>
</tr>
<tr>
<td>RETENTION</td>
<td>SMALLINT</td>
<td>Retention period in days. Null field if not yet collected.</td>
</tr>
<tr>
<td>RETENTION_DATE</td>
<td>INTEGER</td>
<td>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.</td>
</tr>
<tr>
<td>COMPLETE</td>
<td>CHAR(1)</td>
<td>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 it collected with failure.</td>
</tr>
</tbody>
</table>

## DRLLOGDATASETS

This system table contains one row for each collected log data set.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the log definition.</td>
</tr>
<tr>
<td>FIRST_RECORD</td>
<td>VARCHAR(80)</td>
<td>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 <a href="https://example.com">Language Guide and Reference, SH19-6817</a>.</td>
</tr>
<tr>
<td>DATASET_NAME</td>
<td>VARCHAR(54)</td>
<td>Name of the data set, including the member name (for a PDS member).</td>
</tr>
</tbody>
</table>
### Log collector system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPLETE</td>
<td>CHAR(1)</td>
<td>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).</td>
</tr>
<tr>
<td>ELAPSED_SECONDS</td>
<td>INTEGER</td>
<td>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.</td>
</tr>
<tr>
<td>FIRST_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Timestamp of the first record in the log. This is only set if TIMESTAMP expression is specified for the log.</td>
</tr>
<tr>
<td>LOG_SOURCE</td>
<td>CHAR(16)</td>
<td>Reserved.</td>
</tr>
<tr>
<td>LAST_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>Timestamp of the last record in the log. This is only set if TIMESTAMP expression is specified for the log.</td>
</tr>
<tr>
<td>NCOLLECTS</td>
<td>SMALLINT</td>
<td>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.</td>
</tr>
<tr>
<td>NRECORDS</td>
<td>INTEGER</td>
<td>Number of records read from the log data set.</td>
</tr>
<tr>
<td>NSELECTED</td>
<td>INTEGER</td>
<td>Number of records identified.</td>
</tr>
<tr>
<td>NSKIPPED</td>
<td>INTEGER</td>
<td>Number of records skipped due to timestamp overlap (applies when ON TIMESTAMP OVERLAP SKIP specified).</td>
</tr>
<tr>
<td>NUPDATES</td>
<td>INTEGER</td>
<td>Number of database rows updated when the data set was collected.</td>
</tr>
<tr>
<td>NINSERTS</td>
<td>INTEGER</td>
<td>Number of database rows inserted when the data set was collected.</td>
</tr>
<tr>
<td>NDELETES</td>
<td>INTEGER</td>
<td>Number of database rows deleted when the data set was collected.</td>
</tr>
<tr>
<td>RETURN_CODE</td>
<td>SMALLINT</td>
<td>Return code from collect; 0 or 4.</td>
</tr>
<tr>
<td>TIME_COLLECTED</td>
<td>TIMESTAMP</td>
<td>Date and time when collect ended.</td>
</tr>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>ID of the user running collect.</td>
</tr>
<tr>
<td>VOLUME</td>
<td>CHAR(6)</td>
<td>Volume serial number for the data set.</td>
</tr>
</tbody>
</table>

### DRLLOGS

This system table contains one row for each defined log type.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the log.</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.nnn[.APAR_number], where nnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>HEADER</td>
<td>CHAR(1)</td>
<td>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.</td>
</tr>
<tr>
<td>TIMESTAMP_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the TIMESTAMP expression in the DREL EXPRESSIONS table. This is zero if no TIMESTAMP expression is specified.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE_PREFIX</td>
<td>CHAR(8)</td>
<td>Prefix of the table.</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the table.</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.nnn.[APAR_number], where nnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>SQL_CONDITION</td>
<td>VARCHAR(254)</td>
<td>An SQL condition that defines rows to be deleted from the database when the PURGE statement is executed.</td>
</tr>
<tr>
<td>TIME_DEFINED</td>
<td>TIMESTAMP</td>
<td>Date and time when the purge condition was defined.</td>
</tr>
<tr>
<td>CREATOR</td>
<td>CHAR(8)</td>
<td>ID of the user who defined the purge condition.</td>
</tr>
</tbody>
</table>

**DRLRECORDPROCS**

This system table contains one row for each defined record procedure.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM_NAME</td>
<td>K CHAR(8)</td>
<td>Name of the record procedure (name of the load module that gets invoked).</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.nnn.[APAR_number], where nnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>CHAR(8)</td>
<td>Programming language that the record procedure is written in. This is ASM or C.</td>
</tr>
</tbody>
</table>
Log collector system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAMETER_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the PARM expression in the DRLEXPRESSIONS table. This is zero if no PARM expression is specified.</td>
</tr>
<tr>
<td>TIME_DEFINED</td>
<td>TIMESTAMP</td>
<td>Date and time when the record procedure was defined.</td>
</tr>
<tr>
<td>CREATOR</td>
<td>CHAR(8)</td>
<td>ID of the user who defined the record procedure.</td>
</tr>
<tr>
<td>REMARKS</td>
<td>VARCHAR(254)</td>
<td>Description of the record procedure, set by the COMMENT ON statement.</td>
</tr>
</tbody>
</table>

DRLRECORDS
This system table contains one row for each defined record type and one row for each defined header in log definitions.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the record. For a log header, this is &quot;log-name&quot;.</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.nnn[.APAR_number], where nnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>LOG_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the log that contains the record.</td>
</tr>
<tr>
<td>BUILT_BY</td>
<td>CHAR(8)</td>
<td>Name of the record procedure that builds the record, if any.</td>
</tr>
<tr>
<td>NFIELDS</td>
<td>SMALLINT</td>
<td>Number of fields in the record.</td>
</tr>
<tr>
<td>NSECTIONS</td>
<td>SMALLINT</td>
<td>Number of sections in the record.</td>
</tr>
<tr>
<td>CONDITION_NO</td>
<td>SMALLINT</td>
<td>Number of the IDENTIFIED BY condition in the DRLEXPRESSIONS table. This is zero if no IDENTIFIED BY condition is specified.</td>
</tr>
<tr>
<td>TIME_DEFINED</td>
<td>TIMESTAMP</td>
<td>Date and time when the record was defined.</td>
</tr>
<tr>
<td>CREATOR</td>
<td>CHAR(8)</td>
<td>ID of the user who defined the record.</td>
</tr>
<tr>
<td>REMARKS</td>
<td>VARCHAR(254)</td>
<td>Description of the record, set by the COMMENT ON statement.</td>
</tr>
</tbody>
</table>

DRLRPROCINPUT
This system table contains one row for every defined record type that must be processed by a record procedure.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM_NAME</td>
<td>K CHAR(8)</td>
<td>Name of the record procedure.</td>
</tr>
<tr>
<td>RECORD_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the record that is input to the record procedure.</td>
</tr>
</tbody>
</table>

DRLSECTIONS
This system table contains one row for every defined section in defined record types.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the record.</td>
</tr>
<tr>
<td>SECTION_NO</td>
<td>K SMALLINT</td>
<td>Section sequence number within the record.</td>
</tr>
<tr>
<td>SECTION_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the section.</td>
</tr>
</tbody>
</table>
### Log collector system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONDITION_NO</td>
<td>SMALLINT</td>
<td>Number of the PRESENT IF condition in the DRLEXPRESSIONS table. This is zero if no PRESENT IF condition is specified.</td>
</tr>
<tr>
<td>OFFSET_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the OFFSET expression in the DRLEXPRESSIONS table. This is zero if no OFFSET expression is specified.</td>
</tr>
<tr>
<td>LENGTH_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the LENGTH expression in the DRLEXPRESSIONS table. This is zero if no LENGTH expression is specified.</td>
</tr>
<tr>
<td>NUMBER_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the NUMBER expression in the DRLEXPRESSIONS table. This is zero if no NUMBER expression is specified.</td>
</tr>
<tr>
<td>INSECTION_NO</td>
<td>SMALLINT</td>
<td>Number of the section that this section is contained in. This is zero if the section is not contained in another section.</td>
</tr>
<tr>
<td>REPEATED</td>
<td>CHAR(1)</td>
<td>Shows whether the section is repeated. This is Y (the section is repeated) or N (the section is not repeated).</td>
</tr>
</tbody>
</table>

#### DRLUPDATECOLS

This system table contains one row for every column in each update definition, including GROUP BY, SET, and MERGE columns.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATE_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the update definition.</td>
</tr>
<tr>
<td>UPDATECOL_NO</td>
<td>SMALLINT</td>
<td>Sequence number of the column in the update definition.</td>
</tr>
<tr>
<td>COLUMN_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the column.</td>
</tr>
<tr>
<td>COLUMN_NO</td>
<td>SMALLINT</td>
<td>Number of the column in the table definition.</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>EXPRESSION_NO</td>
<td>SMALLINT</td>
<td>Number of the expression in the DRLEXPRESSIONS table.</td>
</tr>
<tr>
<td>COUNT_COLUMN</td>
<td>VARCHAR(18)</td>
<td>If the function is AVG or PERCENT, this contains the name of the column that contains the count of values.</td>
</tr>
<tr>
<td>PERCENTILE</td>
<td>SMALLINT</td>
<td>If the function is PERCENT, this contains the percentile value (1 - 99).</td>
</tr>
</tbody>
</table>

#### DRLUPDATEDISTR

This system table contains one row for every distributed field or column in each update definition.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATE_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the update definition.</td>
</tr>
<tr>
<td>DISTR_NO</td>
<td>SMALLINT</td>
<td>Field or column sequence number in the DISTRIBUTE clause.</td>
</tr>
<tr>
<td>FIELD_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the field or column to be distributed.</td>
</tr>
</tbody>
</table>

#### 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

This system table contains one row for each update definition.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPDATE_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the update definition.</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.mnn[.APAR_number], where mnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>SOURCE_PREFIX</td>
<td>CHAR(8)</td>
<td>Prefix of the source table. This is blank if the source is a record.</td>
</tr>
<tr>
<td>SOURCE_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the source. This is a record name or a table name.</td>
</tr>
<tr>
<td>TARGET_PREFIX</td>
<td>CHAR(8)</td>
<td>Prefix of the target table.</td>
</tr>
<tr>
<td>TARGET_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the target table.</td>
</tr>
<tr>
<td>SECTION_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the repeated section, if any, that is used in the update definition.</td>
</tr>
<tr>
<td>CONDITION_NO</td>
<td>SMALLINT</td>
<td>Number of the WHERE condition in the DRLEXPRESSIONS table. This is zero if no WHERE condition is specified.</td>
</tr>
<tr>
<td>NLETS</td>
<td>SMALLINT</td>
<td>Number of identifiers specified in the LET clause.</td>
</tr>
<tr>
<td>NUPDATECOLS</td>
<td>SMALLINT</td>
<td>Number of columns in the GROUP BY, SET, and MERGE clauses.</td>
</tr>
<tr>
<td>SCHEDULE_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the APPLY SCHEDULE expression in the DRLEXPRESSIONS table. This is zero if APPLY SCHEDULE is not specified.</td>
</tr>
<tr>
<td>SCHEDULE_INTTYPE</td>
<td>VARCHAR(18)</td>
<td>Name of the source column or field that defines the interval type.</td>
</tr>
<tr>
<td>SCHEDULE_START</td>
<td>VARCHAR(18)</td>
<td>Name of the source column or field that defines the interval start timestamp.</td>
</tr>
<tr>
<td>SCHEDULE_END</td>
<td>VARCHAR(18)</td>
<td>Name of the source column or field that defines the interval end time stamp.</td>
</tr>
<tr>
<td>SCHEDULE_STATUS</td>
<td>VARCHAR(18)</td>
<td>Name of the identifier that contains the schedule status.</td>
</tr>
<tr>
<td>NDISTR_FIELDS</td>
<td>SMALLINT</td>
<td>Number of fields or columns that are distributed.</td>
</tr>
<tr>
<td>DISTR_BY_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the Distribute By expression in the DRLEXPRESSIONS table. This is zero if Distribute is not specified.</td>
</tr>
<tr>
<td>DISTR_FROM_EXPR_NO</td>
<td>SMALLINT</td>
<td>Number of the Distribute From expression in the DRLEXPRESSIONS table. This is zero if Distribute is not specified.</td>
</tr>
</tbody>
</table>
**Column name** | **Data type** | **Description**
---|---|---
DISTR_TO_EXPR_NO | SMALLINT | Number of the DISTIBUTE TO expression in the DRLEXPRESSIONS table. This is zero if DISTIBUTE 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFILE</td>
<td>VARCHAR(18)</td>
<td>Profile name, this value is specified on the PROFILE parameter of the GENERATE TABLESPACE and GENERATE INDEX statements.</td>
</tr>
<tr>
<td>COMPONENT_ID</td>
<td>VARCHAR(18)</td>
<td>Tivoli Decision Support for z/OS Component ID or %. Allows for a unique profile for a Tivoli Decision Support for z/OS Component.</td>
</tr>
<tr>
<td>SUBCOMPONENT_ID</td>
<td>VARCHAR(18)</td>
<td>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.</td>
</tr>
<tr>
<td>TABLESPACE_NAME</td>
<td>VARCHAR(18)</td>
<td>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.</td>
</tr>
<tr>
<td>TABLESPACE_TYPE</td>
<td>VARCHAR(9)</td>
<td>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.</td>
</tr>
<tr>
<td>MAXPARTS</td>
<td>INTEGER</td>
<td>Maximum partitions. Used for table space type Growth or Segmented.</td>
</tr>
<tr>
<td>NUMPARTS</td>
<td>INTEGER</td>
<td>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).</td>
</tr>
<tr>
<td>SEGSIZE</td>
<td>INTEGER</td>
<td>Segment size for all table space types.</td>
</tr>
</tbody>
</table>
### Log collector system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBSPACE1</td>
<td>VARCHAR(250)</td>
<td>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).</td>
</tr>
<tr>
<td>TBSPACE2</td>
<td>VARCHAR(250)</td>
<td>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).</td>
</tr>
<tr>
<td>INDEX1</td>
<td>VARCHAR(250)</td>
<td>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').</td>
</tr>
<tr>
<td>INDEX2</td>
<td>VARCHAR(250)</td>
<td>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').</td>
</tr>
</tbody>
</table>

### GENERATE_KEYS

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFILE</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td>COMPONENT_ID</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td>SUBCOMPONENT_ID</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td>TABSPACE_NAME</td>
<td>K</td>
<td>VARCHAR(18)</td>
</tr>
<tr>
<td>PART_NUM</td>
<td>K</td>
<td>INTEGER</td>
</tr>
<tr>
<td>PARTITION_OPTIONS</td>
<td>VARCHAR(250)</td>
<td></td>
</tr>
<tr>
<td>RANGE_COLUMNS</td>
<td>VARCHAR(250)</td>
<td>Specifies the columns of the key.</td>
</tr>
<tr>
<td>PARTITION_KEY</td>
<td>VARCHAR(250)</td>
<td>The limit key for the partition boundary.</td>
</tr>
</tbody>
</table>
## 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHART_NAME</td>
<td>CHAR(8)</td>
<td>ADMCFORM name. This is the same as the CHART column in the DRLREPORTS table.</td>
</tr>
<tr>
<td>TYPE</td>
<td>SMALLINT</td>
<td>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.</td>
</tr>
<tr>
<td>VALUES</td>
<td>SMALLINT</td>
<td>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</td>
</tr>
<tr>
<td>AXIS_ORIENTATION</td>
<td>SMALLINT</td>
<td>Axis orientation. This can be 1 or 2. 1 means vertical y-axis and bars. 2 means horizontal y-axis and bars.</td>
</tr>
</tbody>
</table>
# Dialog system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y_DATA_TYPE</td>
<td>VARCHAR(50)</td>
<td>If the chart type is 10 (combination), this column shows the chart type for each data group:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Line chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Surface chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Histogram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41, 42, 43 Bar chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, 1, 42, 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.</td>
</tr>
<tr>
<td>X_AXIS_TITLE</td>
<td>VARCHAR(52)</td>
<td>This is a string containing the x-axis title.</td>
</tr>
<tr>
<td>Y_AXIS_TITLE</td>
<td>VARCHAR(52)</td>
<td>This is a string containing the y-axis title.</td>
</tr>
</tbody>
</table>

## DRLCOMPONENTS

This system table contains one row for each Tivoli Decision Support for z/OS component.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONENT_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the component.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VARCHAR(50)</td>
<td>Description of the component that is shown in the dialog.</td>
</tr>
<tr>
<td>STATUS</td>
<td>CHAR(1)</td>
<td>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.</td>
</tr>
<tr>
<td>TIME_INSTALLED</td>
<td>TIMESTAMP</td>
<td>Date and time when the component was installed or defined.</td>
</tr>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>ID of the user who installed or defined the component.</td>
</tr>
</tbody>
</table>

## DRLCOMP_OBJECTS

This system table contains one row for every object in each component.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONENT_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the component.</td>
</tr>
<tr>
<td>OBJECT_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the object.</td>
</tr>
<tr>
<td>OBJECT_TYPE</td>
<td>CHAR(8)</td>
<td>Type of object. This is LOG, RECORD, RECPROC, TABSPACE, LOOKUP, TABLE, UPDATE, REPORT, or REPGROUP.</td>
</tr>
<tr>
<td>MEMBER_NAME</td>
<td>CHAR(8)</td>
<td>Name of the member in the SDRLDEFS or SDRLRxxx library where the object is defined.</td>
</tr>
<tr>
<td>PART_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the component part that the object belongs to, if any.</td>
</tr>
<tr>
<td>EXCLUDE_FLAG</td>
<td>CHAR(1)</td>
<td>Flag to determine if this object is excluded from installation of the component.</td>
</tr>
</tbody>
</table>

## DRLCOMP_PARTS

This system table contains one row for every part in each component.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONENT_NAME</td>
<td>VARCHAR(18)</td>
<td>Name of the component.</td>
</tr>
</tbody>
</table>
### Dialog system tables

**PART_NAME**
- **Data type**: VARCHAR(18)
- **Description**: Name of the component part.

**DESCRIPTION**
- **Data type**: VARCHAR(50)
- **Description**: Description of the component part that is shown in the dialog.

**STATUS**
- **Data type**: CHAR(1)
- **Description**: 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**
- **Data type**: TIMESTAMP
- **Description**: Date and time when the component part was installed or defined.

**USER_ID**
- **Data type**: CHAR(8)
- **Description**: ID of the user who installed or defined the component part.

---

**DRLGROUPS**

This system table contains one row for each defined report group.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP_NAME</td>
<td>K</td>
<td>Group ID.</td>
</tr>
<tr>
<td>GROUP_OWNER</td>
<td>K</td>
<td>Owner of the group. This is blank for a public group.</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.nnn.APAR_number, where nnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VARCHAR(50)</td>
<td>Description of the group that is shown in the dialog.</td>
</tr>
<tr>
<td>TIME_CREATED</td>
<td>TIMESTAMP</td>
<td>Date and time when the group was defined.</td>
</tr>
<tr>
<td>CREATOR</td>
<td>CHAR(8)</td>
<td>ID of the user who defined the group.</td>
</tr>
</tbody>
</table>

---

**DRLGROUP_REPORTS**

This system table contains one row for every report in each defined report group.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP_NAME</td>
<td>K</td>
<td>Group ID.</td>
</tr>
<tr>
<td>GROUP_OWNER</td>
<td>K</td>
<td>Owner of the group.</td>
</tr>
<tr>
<td>REPORT_NAME</td>
<td>K</td>
<td>ID of the report that belongs to the group.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>K</td>
<td>Owner of the report that belongs to the group.</td>
</tr>
</tbody>
</table>

---

**DRLREPORTS**

This system table contains one row for each defined report.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT_NAME</td>
<td>K</td>
<td>Report ID.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>K</td>
<td>Owner of the report. This is blank for a public report.</td>
</tr>
<tr>
<td>VERSION</td>
<td>VARCHAR(18)</td>
<td>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.nnn.APAR_number, where nnn is the version, release, modification level of the object.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>VARCHAR(50)</td>
<td>Description of the report that is shown in the dialog.</td>
</tr>
</tbody>
</table>
## Dialog system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>CHAR(8)</td>
<td>Type of report. This is QUERY, TABDATA, or GRAPH.</td>
</tr>
<tr>
<td>BATCH</td>
<td>CHAR(1)</td>
<td>Y if the report should be produced in batch; N otherwise.</td>
</tr>
<tr>
<td>PRINT</td>
<td>CHAR(1)</td>
<td>Y if the report should be printed when produced in batch; N otherwise.</td>
</tr>
<tr>
<td>SAVE</td>
<td>CHAR(1)</td>
<td>Y if the report should be saved when produced in batch; N otherwise.</td>
</tr>
<tr>
<td>RUN_CYCLE</td>
<td>CHAR(8)</td>
<td>Batch run cycle for the report. This is DAILY, WEEKLY, or MONTHLY.</td>
</tr>
<tr>
<td>QUERY_PREFIX</td>
<td>CHAR(8)</td>
<td>Prefix of the QMF query that should be run when the report is produced.</td>
</tr>
<tr>
<td>QUERY</td>
<td>VARCHAR(18)</td>
<td>Name of the QMF query that should be run when the report is produced.</td>
</tr>
<tr>
<td>FORM_PREFIX</td>
<td>CHAR(8)</td>
<td>Prefix of the QMF form that should be used when the report is produced.</td>
</tr>
<tr>
<td>FORM</td>
<td>VARCHAR(18)</td>
<td>Name of the QMF form that should be used when the report is produced.</td>
</tr>
<tr>
<td>CHART</td>
<td>CHAR(8)</td>
<td>Name of the GDDM-ICU format to be used for the report. Blank means that the report is tabular.</td>
</tr>
<tr>
<td>FILE</td>
<td>CHAR(8)</td>
<td>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).</td>
</tr>
<tr>
<td>MACRO</td>
<td>CHAR(8)</td>
<td>Not used.</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>VARCHAR(254)</td>
<td>Name of the table or tables on which the the report is bases. This is extracted from the query when the report is defined.</td>
</tr>
<tr>
<td>NVARIABLES</td>
<td>SMALLINT</td>
<td>Number of variables defined for the report or extracted from the query.</td>
</tr>
<tr>
<td>NATTRIBUTES</td>
<td>SMALLINT</td>
<td>Number of attributes defined for the report.</td>
</tr>
<tr>
<td>TIME_CREATED</td>
<td>TIMESTAMP</td>
<td>Date and time when the report was defined.</td>
</tr>
<tr>
<td>CREATOR</td>
<td>CHAR(8)</td>
<td>ID of the user who defined the report.</td>
</tr>
<tr>
<td>REMARKS</td>
<td>VARCHAR(254)</td>
<td>Long free-format description of the report that can be entered from the dialog.</td>
</tr>
<tr>
<td>FINAL_SUMMARY</td>
<td>CHAR(3)</td>
<td>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.</td>
</tr>
<tr>
<td>ACROSS_SUMMARY</td>
<td>CHAR(3)</td>
<td>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.</td>
</tr>
</tbody>
</table>

### DRLREPORT_ATTR

This system table contains one row for every attribute in each defined report.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT_NAME</td>
<td>K VARCHAR(18)</td>
<td>Report ID.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>CHAR(8)</td>
<td>Owner of the report. This is blank for a public report.</td>
</tr>
<tr>
<td>ATTRIBUTE_NO</td>
<td>K SMALLINT</td>
<td>Attribute sequence number.</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>VARCHAR(18)</td>
<td>Attribute value.</td>
</tr>
</tbody>
</table>
**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT_NAME</td>
<td>K VARCHAR(18)</td>
<td>Report ID.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>K CHAR(8)</td>
<td>Owner of the report. This is blank for a public report.</td>
</tr>
<tr>
<td>COLUMN_NO</td>
<td>K SMALLINT</td>
<td>Column number.</td>
</tr>
<tr>
<td>HEADING</td>
<td>VARCHAR(40)</td>
<td>Column heading.</td>
</tr>
<tr>
<td>USAGE</td>
<td>CHAR(7)</td>
<td>Usage code.</td>
</tr>
<tr>
<td>INDENT</td>
<td>SMALLINT</td>
<td>Column indentation.</td>
</tr>
<tr>
<td>WIDTH</td>
<td>SMALLINT</td>
<td>Column width.</td>
</tr>
<tr>
<td>EDIT</td>
<td>CHAR(5)</td>
<td>Edit code.</td>
</tr>
<tr>
<td>SEQ</td>
<td>SMALLINT</td>
<td>Column sequence number.</td>
</tr>
<tr>
<td>DEFINITION</td>
<td>VARCHAR(50)</td>
<td>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 &amp;n type, where n 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.</td>
</tr>
</tbody>
</table>

**DRLREPORT_QUERIES**
This system table contains one row for every query line in each defined report, if QMF is not used.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT_NAME</td>
<td>K VARCHAR(18)</td>
<td>Report ID.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>K CHAR(8)</td>
<td>Owner of the report. This is blank for a public report.</td>
</tr>
<tr>
<td>LINE_NO</td>
<td>K SMALLINT</td>
<td>Line number in the query.</td>
</tr>
<tr>
<td>QUERY_LINE</td>
<td>VARCHAR(80)</td>
<td>Query text.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT_NAME</td>
<td>K VARCHAR(18)</td>
<td>Report ID.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>K CHAR(8)</td>
<td>Owner of the report. This is blank for a public report.</td>
</tr>
<tr>
<td>TYPE</td>
<td>K CHAR(8)</td>
<td>Text type. This is HEADING, FOOTING, DETAIL, FINAL or ROWS.</td>
</tr>
<tr>
<td>LINE_NO</td>
<td>K SMALLINT</td>
<td>Line number for HEADING and FOOTING.</td>
</tr>
<tr>
<td>ALIGNMENT</td>
<td>CHAR(6)</td>
<td>Shows how the text should be aligned; left, center, or right.</td>
</tr>
<tr>
<td>TEXT</td>
<td>VARCHAR(55)</td>
<td>Text for one line of a report text (see TYPE above).</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT_NAME</td>
<td>K VARCHAR(18)</td>
<td>Report ID.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>K CHAR(8)</td>
<td>Owner of the report. This is blank for a public report.</td>
</tr>
<tr>
<td>VARIABLE_NO</td>
<td>K SMALLINT</td>
<td>Sequence number of the variable.</td>
</tr>
<tr>
<td>VARIABLE_NAME</td>
<td>V VARCHAR(18)</td>
<td>Name of the variable.</td>
</tr>
<tr>
<td>EXPRESSION</td>
<td>V VARCHAR(80)</td>
<td>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.</td>
</tr>
<tr>
<td>OPERATOR</td>
<td>CHAR(4)</td>
<td>Operator that is used when comparing the variable and the expression, if the variable is found in the query. This is =, &lt;=, &gt;=, IN, or LIKE.</td>
</tr>
<tr>
<td>DATA_TYPE</td>
<td>CHAR(8)</td>
<td>Data type of the variable, if specified. This is CHAR, NUMERIC, DATE, TIME, or TIMESTAMP.</td>
</tr>
<tr>
<td>REQUIRED</td>
<td>CHAR(1)</td>
<td>Shows whether the variable must be given a value. This is Y for yes, or N or blank for no.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>V VARCHAR(40)</td>
<td>Default value to use for the variable, if specified.</td>
</tr>
<tr>
<td>IN_HEADER</td>
<td>CHAR(1)</td>
<td>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.</td>
</tr>
<tr>
<td>IN_HEADER_VALUE</td>
<td>V VARCHAR(35)</td>
<td>Default header value for a non-required variable without a substitution value.</td>
</tr>
</tbody>
</table>

DRLSEARCH_ATTR

This system table contains one row for every attribute in each saved report search.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the saved search.</td>
</tr>
<tr>
<td>SEARCH_OWNER</td>
<td>K CHAR(8)</td>
<td>Owner of the saved search. This is blank for a public search.</td>
</tr>
<tr>
<td>ATTR_SET_NO</td>
<td>K SMALLINT</td>
<td>Attribute set sequence number. The attribute sets are logically ORed together.</td>
</tr>
<tr>
<td>ATTRIBUTE_NO</td>
<td>K SMALLINT</td>
<td>Attribute sequence number within the attribute set. The attributes within a set are logically ANDeD together.</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>V VARCHAR(18)</td>
<td>Attribute value. This can contain global search characters.</td>
</tr>
</tbody>
</table>

DRLSEARCHES

This system table contains one row for each saved report search.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEARCH_NAME</td>
<td>K VARCHAR(18)</td>
<td>Name of the saved search.</td>
</tr>
<tr>
<td>SEARCH_OWNER</td>
<td>K CHAR(8)</td>
<td>Owner of the saved search. This is blank for a public search.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>V VARCHAR(50)</td>
<td>Description of the search that is shown in the dialog.</td>
</tr>
</tbody>
</table>
### Dialog system tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NATTR_SETS</td>
<td>SMALLINT</td>
<td>Number of attribute sets used in the search.</td>
</tr>
<tr>
<td>REPORT_DESC</td>
<td>VARCHAR(50)</td>
<td>Report description used in the search. This can contain global search characters.</td>
</tr>
<tr>
<td>REPORT_TYPE</td>
<td>CHAR(8)</td>
<td>Report type specified in the search.</td>
</tr>
<tr>
<td>REPORT_OWNER</td>
<td>CHAR(8)</td>
<td>Report owner specified in the search.</td>
</tr>
<tr>
<td>TIME_CREATED</td>
<td>TIMESTAMP</td>
<td>Date and time when the search was saved.</td>
</tr>
<tr>
<td>CREATOR</td>
<td>CHAR(8)</td>
<td>ID of the user who saved the search.</td>
</tr>
</tbody>
</table>

### Views on DB2 and QMF tables

These views on DB2 tables are required for users without access to the tables.

<table>
<thead>
<tr>
<th>View name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLCOLUMNS</td>
<td>This view is based on SYSIBM.SYSCOLUMNS in the DB2 catalog. It is used to get column names and comments.</td>
</tr>
<tr>
<td>DRLINDEXES</td>
<td>This table is based on SYSIBM.SYSINDEXES in the DB2 catalog. It is used to get table index information.</td>
</tr>
<tr>
<td>DRLINDEXPART</td>
<td>This view is based on SYSIBM.SYSINDEXPART in the DB2 catalog. It is used to get index partition information.</td>
</tr>
<tr>
<td>DRLKEYS</td>
<td>This view is based on SYSIBM.SYSKEYS in the DB2 catalog. It is used to get information on index keys.</td>
</tr>
<tr>
<td>DRLOBJEC_DATA</td>
<td>This view is based on Q.OBJECT_DATA, a QMF control table that contains information about QMF objects.</td>
</tr>
<tr>
<td>DRLTABLEAUTH</td>
<td>This view is based on SYSIBM.SYSTABLEAUTH in the DB2 catalog. It is used to get table privilege information.</td>
</tr>
<tr>
<td>DRLTABLEPART</td>
<td>This view is based on SYSIBM.SYSTABLEPART in the DB2 catalog. It is used to get tablespace information.</td>
</tr>
<tr>
<td>DRLTABLES</td>
<td>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.</td>
</tr>
<tr>
<td>DRLTABLESPACE</td>
<td>This view is based on SYSIBM.SYSTABLESPACE in the DB2 catalog. It is used to get a list of table spaces.</td>
</tr>
<tr>
<td>DRLVIEWS</td>
<td>This view is based on SYSIBM.SYSVIEWS in the DB2 catalog. It is used to get view definitions.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>View Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLUSER_GROUPREPS</td>
<td>This view is based on DRLGROUP_REPORTS. It allows a user to update only his own report groups.</td>
</tr>
<tr>
<td>DRLUSER_GROUPS</td>
<td>This view is based on DRLGROUPS. It allows a user to update only his own report groups.</td>
</tr>
<tr>
<td>DRLUSER_REPORTATTR</td>
<td>This view is based on DRLREPORT_ATTR. It allows a user to update only his own reports.</td>
</tr>
<tr>
<td>View Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DRLUSER_REPORTS</td>
<td>This view is based on DRLREPORTS. It allows a user to update only his own</td>
</tr>
<tr>
<td></td>
<td>reports.</td>
</tr>
<tr>
<td>DRLUSER_REPORTVARS</td>
<td>This view is based on DRLREPORT_VARS. It allows a user to update only his</td>
</tr>
<tr>
<td></td>
<td>own reports.</td>
</tr>
<tr>
<td>DRLUSER_SEARCHATTR</td>
<td>This view is based on DRLSEARCH_ATTR. It allows a user to update only his</td>
</tr>
<tr>
<td></td>
<td>own searches.</td>
</tr>
<tr>
<td>DRLUSER_SEARCHES</td>
<td>This view is based on DRLSEARCHES. It allows a user to update only his own</td>
</tr>
<tr>
<td></td>
<td>searches.</td>
</tr>
<tr>
<td>DRLUSER_REPORTQRY</td>
<td>This view is based on DRLREPORT_ QUERIES. It allows a user to update only</td>
</tr>
<tr>
<td></td>
<td>his own reports.</td>
</tr>
<tr>
<td>DRLUSER_REPORTCOLS</td>
<td>This view is based on DRLREPORT_COLUMN S. It allows a user to update only his</td>
</tr>
<tr>
<td></td>
<td>own reports.</td>
</tr>
<tr>
<td>DRLUSER REPORTTEXT</td>
<td>This view is based on DRLREPORT_TEXT. It allows a user to update only his</td>
</tr>
<tr>
<td></td>
<td>own reports.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>TABLE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABILITY_D</td>
</tr>
<tr>
<td>AVAILABILITY_W</td>
</tr>
<tr>
<td>AVAILABILITY_M</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY_OF_WEEK</td>
<td>K</td>
<td>SMALLINT Day of week number, 1 through 7 (Monday through Sunday).</td>
</tr>
<tr>
<td>DAY_TYPE</td>
<td>CHAR(8)</td>
<td>Day type for the day of week.</td>
</tr>
</tbody>
</table>

**Example of table contents**

<table>
<thead>
<tr>
<th>DAY OF DAY WEEK TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>----</td>
</tr>
</tbody>
</table>

243
Control tables

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERIOD_PLAN_ID</td>
<td>K</td>
<td>CHAR(8) You can have different sets of period names for different systems.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each application normally uses a system ID from the log to match this field,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for example the MVS system ID for an MVS performance application.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify % for the rows that specify your default set of period names.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This can contain global search characters.</td>
</tr>
<tr>
<td>DAY_TYPE</td>
<td>K</td>
<td>CHAR(8) Day type the period applies to. This can be any of the day types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>specified in the DAY_OF_WEEK and SPECIAL_DAY control tables.</td>
</tr>
<tr>
<td>START_TIME</td>
<td>K</td>
<td>TIME Time when the period starts.</td>
</tr>
<tr>
<td>END_TIME</td>
<td></td>
<td>TIME Time when the period ends.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td></td>
<td>CHAR(8) Name of the period.</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>PERIOD_PLAN</th>
<th>DAY_TYPE</th>
<th>START_TIME</th>
<th>END_TIME</th>
<th>PERIOD_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>TYPE</td>
<td>TIME</td>
<td>TIME</td>
<td>NAME</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>%</td>
<td>MON</td>
<td>00.00.00</td>
<td>08.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>MON</td>
<td>08.00.00</td>
<td>17.00.00</td>
<td>PRIME</td>
</tr>
<tr>
<td>%</td>
<td>MON</td>
<td>17.00.00</td>
<td>24.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>TUE</td>
<td>00.00.00</td>
<td>08.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>TUE</td>
<td>08.00.00</td>
<td>17.00.00</td>
<td>PRIME</td>
</tr>
<tr>
<td>%</td>
<td>TUE</td>
<td>17.00.00</td>
<td>24.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>WED</td>
<td>00.00.00</td>
<td>08.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>WED</td>
<td>08.00.00</td>
<td>17.00.00</td>
<td>PRIME</td>
</tr>
<tr>
<td>%</td>
<td>WED</td>
<td>17.00.00</td>
<td>24.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>THU</td>
<td>00.00.00</td>
<td>08.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>THU</td>
<td>08.00.00</td>
<td>17.00.00</td>
<td>PRIME</td>
</tr>
<tr>
<td>%</td>
<td>THU</td>
<td>17.00.00</td>
<td>24.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>FRI</td>
<td>00.00.00</td>
<td>08.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>FRI</td>
<td>08.00.00</td>
<td>17.00.00</td>
<td>PRIME</td>
</tr>
<tr>
<td>%</td>
<td>FRI</td>
<td>17.00.00</td>
<td>24.00.00</td>
<td>NIGHT</td>
</tr>
<tr>
<td>%</td>
<td>SAT</td>
<td>00.00.00</td>
<td>24.00.00</td>
<td>WEEKEND</td>
</tr>
<tr>
<td>%</td>
<td>SUN</td>
<td>00.00.00</td>
<td>24.00.00</td>
<td>WEEKEND</td>
</tr>
<tr>
<td>%</td>
<td>HOLIDAY</td>
<td>00.00.00</td>
<td>24.00.00</td>
<td>HOLIDAY</td>
</tr>
</tbody>
</table>

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.
**Control tables**

### 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

<table>
<thead>
<tr>
<th>SCHEDULE_NAME</th>
<th>DAY</th>
<th>START</th>
<th>END</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANDARD</td>
<td>MON</td>
<td>08.00.00</td>
<td>17.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>TUE</td>
<td>08.00.00</td>
<td>17.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>WED</td>
<td>08.00.00</td>
<td>17.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>THU</td>
<td>08.00.00</td>
<td>17.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>FRI</td>
<td>08.00.00</td>
<td>17.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>SAT</td>
<td>00.00.00</td>
<td>00.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>SUN</td>
<td>00.00.00</td>
<td>00.00.00</td>
</tr>
<tr>
<td>STANDARD</td>
<td>HOLIDAY</td>
<td>00.00.00</td>
<td>00.00.00</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K DATE</td>
<td>Date to be defined as special day.</td>
</tr>
<tr>
<td>DAY_TYPE</td>
<td>CHAR(8)</td>
<td>Day type for the date; for example, HOLIDAY.</td>
</tr>
</tbody>
</table>

### Example of table contents

<table>
<thead>
<tr>
<th>DAY</th>
<th>DATE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1999-12-25</td>
<td>HOLIDAY</td>
</tr>
<tr>
<td></td>
<td>2000-01-01</td>
<td>HOLIDAY</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGGR_TABLE</td>
<td>K CHAR(18)</td>
<td>Name of TDS table.</td>
</tr>
<tr>
<td>AGGR_COLUMN</td>
<td>K CHAR(18)</td>
<td>Name of TDS column.</td>
</tr>
<tr>
<td>AGGR_DEF_VALUE</td>
<td>CHAR(16)</td>
<td>Default value to assign to field.</td>
</tr>
</tbody>
</table>
Example of table contents

<table>
<thead>
<tr>
<th>AGGR TABLE</th>
<th>AGGR COLUMN</th>
<th>AGGR DEF VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2_PACKAGE_H</td>
<td>CORRELATION_ID</td>
<td>$USER</td>
</tr>
<tr>
<td>DB2_PACKAGE_H</td>
<td>PRIMARY_AUTH_ID</td>
<td>$USER</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>K CHAR(4)</td>
<td>MVS system ID. From SMFMNSID (V3) or SMFSID (V2).</td>
</tr>
<tr>
<td>CICS_SYSTEM_ID</td>
<td>K CHAR(8)</td>
<td>CICS generic ID. This is the VTAM® application identifier for the CICS system that produced the dictionary. From SMFMNPRN (V3) or SMFPSPRN (V2).</td>
</tr>
<tr>
<td>CLASS</td>
<td>K SMALLINT</td>
<td>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).</td>
</tr>
<tr>
<td>VERSION</td>
<td>K SMALLINT</td>
<td>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).</td>
</tr>
<tr>
<td>FIELD_NO</td>
<td>K SMALLINT</td>
<td>Assigned connector for this dictionary entry (CMODCONN). This is also the index to the dictionary entry array.</td>
</tr>
<tr>
<td>CICS_VER</td>
<td>K CHAR(4)</td>
<td>CICS version and release that created this dictionary (from the field SMFMNRVN). EX. 0410.</td>
</tr>
<tr>
<td>DICT_ENTRY_ID</td>
<td>CHAR(12)</td>
<td>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.</td>
</tr>
<tr>
<td>OUTPUT_LENGTH</td>
<td>SMALLINT</td>
<td>Field length for matching DICT_ENTRY_ID in CICS_FIELD. It is used for building the output record.</td>
</tr>
<tr>
<td>OUTPUT_OFFSET</td>
<td>SMALLINT</td>
<td>Field offset for matching DICT_ENTRY_ID in CICS_FIELD. It is used for building the output record.</td>
</tr>
<tr>
<td>USED</td>
<td>CHAR(8)</td>
<td>A flag indicating (if = Y) that this dictionary entry has been updated with field length and offset data from a matching DICT_ENTRY_ID in CICS_FIELD.</td>
</tr>
</tbody>
</table>

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
Control tables

dictionary fields described in “CICS_DICTIONARY” on page 246. For more information, refer to the CICS Performance Feature Guide and Reference.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS</td>
<td>K SMALLINT</td>
<td>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).</td>
</tr>
<tr>
<td>DICT_ENTRY_ID</td>
<td>K CHAR(12)</td>
<td>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.</td>
</tr>
<tr>
<td>FIRST_CICS_VER</td>
<td>K CHAR(4)</td>
<td>This is the 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</td>
</tr>
<tr>
<td>OUTPUT_LENGTH</td>
<td>SMALLINT</td>
<td>This is the field length that is used to build the output record.</td>
</tr>
<tr>
<td>OUTPUT_OFFSET</td>
<td>INTEGER</td>
<td>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.</td>
</tr>
</tbody>
</table>

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
### AVAILABLEABILITY_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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>System ID such as an MVS or VM system ID.</td>
</tr>
<tr>
<td>AREA</td>
<td>CHAR(8)</td>
<td>Major area the resource is related to, such as MVS or NETWORK.</td>
</tr>
<tr>
<td>RESOURCE_TYPE</td>
<td>CHAR(8)</td>
<td>Resource type.</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>CHAR(8)</td>
<td>Resource name.</td>
</tr>
<tr>
<td>RESOURCE_GROUP</td>
<td>CHAR(8)</td>
<td>Resource group.</td>
</tr>
</tbody>
</table>
### INTER

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>K DATE</td>
<td>Date when the exception occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>K TIME</td>
<td>Time when the exception occurred.</td>
</tr>
<tr>
<td>SYSTEM_ID</td>
<td>K CHAR(8)</td>
<td>System where the exception occurred.</td>
</tr>
<tr>
<td>AREA</td>
<td>K CHAR(8)</td>
<td>Major area the exception is related to, such as MVS or NETWORK.</td>
</tr>
<tr>
<td>EXCEPTION_ID</td>
<td>K VARCHAR(18)</td>
<td>Short description of the exception type. This can be used to count the number of exceptions of different types.</td>
</tr>
<tr>
<td>RESOURCE_NAME1</td>
<td>K CHAR(8)</td>
<td>Name of the first resource that the exception is related to.</td>
</tr>
<tr>
<td>RESOURCE_NAME2</td>
<td>K CHAR(8)</td>
<td>Name of the second resource that the exception is related to.</td>
</tr>
<tr>
<td>DATE_GENERATED</td>
<td>DATE</td>
<td>Date when the problem was recorded in the Information/Management database. This is null if no problem record has been generated.</td>
</tr>
<tr>
<td>EXCEPTION_DESC</td>
<td>VARCHAR(45)</td>
<td>Text that describes the exception, in any format.</td>
</tr>
<tr>
<td>PROBLEM_FLAG</td>
<td>CHAR(1)</td>
<td>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).</td>
</tr>
<tr>
<td>PROBLEM_NUMBER</td>
<td>CHAR(8)</td>
<td>The Information/Management problem-record number. This is null if no problem record has been generated.</td>
</tr>
<tr>
<td>SEVERITY</td>
<td>CHAR(2)</td>
<td>Severity of the problem. This is user-defined.</td>
</tr>
<tr>
<td>TRANSACT_NUMBER</td>
<td>INTEGER</td>
<td>Transaction identifier number.</td>
</tr>
<tr>
<td>TRANSACT_CHAR</td>
<td>CHAR(4)</td>
<td>Transaction number in character format. (in some special cases CICS system tasks are identified as III, JBS, J01-J99, TCB.)</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>CHAR(8)</td>
<td>Name of the program.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_NAME</td>
<td>CHAR(8)</td>
<td>Migration job name.</td>
</tr>
<tr>
<td>STEP_NO</td>
<td>INTEGER</td>
<td>Step number of job.</td>
</tr>
<tr>
<td>START_DATE</td>
<td>DATE</td>
<td>Start date of job.</td>
</tr>
<tr>
<td>START_TIME</td>
<td>TIME</td>
<td>Start time of job.</td>
</tr>
<tr>
<td>STEP_NAME</td>
<td>CHAR(30)</td>
<td>Step name of job.</td>
</tr>
<tr>
<td>RETURN_CODE</td>
<td>INTEGER</td>
<td>Step status code.</td>
</tr>
<tr>
<td>COMPLETED_CODE</td>
<td>CHAR</td>
<td>Y – Completed successfully</td>
</tr>
<tr>
<td>END_DATE</td>
<td>DATE</td>
<td>End date of last migration step.</td>
</tr>
<tr>
<td>END_TIME</td>
<td>TIME</td>
<td>End time of last migration step.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>System ID associated with the resource. This can contain global search characters.</td>
</tr>
<tr>
<td>AREA</td>
<td>CHAR(8)</td>
<td>Major area that the resource is related to, such as MVS or NETWORK. This can contain global search characters.</td>
</tr>
<tr>
<td>RESOURCE_TYPE</td>
<td>CHAR(8)</td>
<td>Resource type. This can contain global search characters.</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>CHAR(8)</td>
<td>Resource name. This can contain global search characters.</td>
</tr>
<tr>
<td>RESOURCE_GROUP</td>
<td>CHAR(8)</td>
<td>Resource group. This can contain global search characters.</td>
</tr>
<tr>
<td>AVAIL_OBJ_PCT</td>
<td>DECIMAL(4,1)</td>
<td>Availability objective for the resource, in percent.</td>
</tr>
<tr>
<td>SCHEDULE_NAME</td>
<td>CHAR(8)</td>
<td>Schedule name to use for the resource.</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>SYSTEM_ID</th>
<th>AREA</th>
<th>RESOURCE_TYPE</th>
<th>RESOURCE_NAME</th>
<th>RESOURCE_GROUP</th>
<th>SCHEDULE_NAME</th>
<th>AVAIL_OBJ_PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------</td>
<td>----</td>
<td>-------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>

250 IBM Tivoli Decision Support for z/OS: Administration Guide and Reference
**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM_ID</td>
<td>K CHAR(8)</td>
<td>System ID such as an MVS or VM system ID. This can contain global search characters.</td>
</tr>
<tr>
<td>SUBSYSTEM_ID</td>
<td>K CHAR(8)</td>
<td>Subsystem ID such as TSO or a CICS* system ID. This can contain global search characters. This is not used in the predefined tables.</td>
</tr>
<tr>
<td>USER_ID</td>
<td>K CHAR(8)</td>
<td>User ID of the user to be grouped. This can contain global search characters.</td>
</tr>
<tr>
<td>DEPARTMENT</td>
<td>CHAR(8)</td>
<td>Department that the user belongs to. This is not used in the predefined tables.</td>
</tr>
<tr>
<td>DIVISION</td>
<td>CHAR(8)</td>
<td>Division that the user belongs to. This is not used in the predefined tables.</td>
</tr>
<tr>
<td>GROUP_NAME</td>
<td>CHAR(8)</td>
<td>Name of the group that the user belongs to.</td>
</tr>
</tbody>
</table>

**Example of table contents**

<table>
<thead>
<tr>
<th>SYSTEM_ID</th>
<th>SUBSYSTEM_ID</th>
<th>USER_ID</th>
<th>DIVISION</th>
<th>DEPARTMENT</th>
<th>GROUP_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>USER1</td>
<td></td>
<td></td>
<td>GROUP1</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>USER2</td>
<td></td>
<td></td>
<td>GROUP2</td>
</tr>
</tbody>
</table>

...
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.

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date. For the _M table, this is the date of the first day of the month. From S01DATE.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>Time rounded down to the nearest hour. This applies only to the _H table. From S01TIME.</td>
</tr>
<tr>
<td>SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>System ID. From S01SYST.</td>
</tr>
<tr>
<td>DEPARTMENT_NAME</td>
<td>CHAR(8)</td>
<td>Department name. From DEPARTMENT_NAME in the SAMPLE_USER lookup table. This is derived using field S01USER from the record as key.</td>
</tr>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>User ID. From S01USER.</td>
</tr>
<tr>
<td>CPU_SECONDS</td>
<td>FLOAT</td>
<td>Total processor time, in seconds. Calculated as the sum of S01CPU/100.0.</td>
</tr>
<tr>
<td>PAGES_PRINTED</td>
<td>INTEGER</td>
<td>Number of pages printed. This is the sum of S01PRNT.</td>
</tr>
<tr>
<td>RESPONSE_SECONDS</td>
<td>INTEGER</td>
<td>Total response time, in seconds. This is the sum of S01RESP.</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>INTEGER</td>
<td>Number of transactions. This is the sum of S01TRNS.</td>
</tr>
</tbody>
</table>

SAMPLE_USER lookup table

This lookup table assigns department names to users.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>User ID</td>
</tr>
<tr>
<td>DEPARTMENT_NAME</td>
<td>CHAR(8)</td>
<td>Department name</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>USER_ID</th>
<th>DEPARTMENT_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAMS</td>
<td>Appl Dev</td>
</tr>
<tr>
<td>GEYER</td>
<td>Finance</td>
</tr>
<tr>
<td>GOUNOT</td>
<td>Retail</td>
</tr>
<tr>
<td>HAAS</td>
<td>Finance</td>
</tr>
<tr>
<td>JONES</td>
<td>Appl Dev</td>
</tr>
<tr>
<td>KWAN</td>
<td>Marketng</td>
</tr>
<tr>
<td>LEE</td>
<td>Manufact</td>
</tr>
<tr>
<td>LUTZ</td>
<td>Manufact</td>
</tr>
<tr>
<td>MARINO</td>
<td>Retail</td>
</tr>
<tr>
<td>MEHTA</td>
<td>Manufact</td>
</tr>
<tr>
<td>PARKER</td>
<td>Finance</td>
</tr>
<tr>
<td>PEREZ</td>
<td>Retail</td>
</tr>
</tbody>
</table>

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
**Sample Report 1**

![Sample Report 1](image)

*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

<table>
<thead>
<tr>
<th>Month start date</th>
<th>Department name</th>
<th>User ID</th>
<th>Transactions</th>
<th>Average response seconds</th>
<th>CPU seconds</th>
<th>Pages printed</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000-01-01</td>
<td>Appl Dev</td>
<td>ADAMS</td>
<td>1109</td>
<td>3.84</td>
<td>244.13</td>
<td>821</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JONES</td>
<td>1138</td>
<td>3.40</td>
<td>228.79</td>
<td>1055</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMITH</td>
<td>870</td>
<td>4.27</td>
<td>183.03</td>
<td>864</td>
</tr>
<tr>
<td></td>
<td>Finance</td>
<td>Geyer</td>
<td>509</td>
<td>4.29</td>
<td>115.97</td>
<td>529</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Has</td>
<td>786</td>
<td>3.56</td>
<td>137.48</td>
<td>648</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parker</td>
<td>462</td>
<td>6.79</td>
<td>171.51</td>
<td>704</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spencer</td>
<td>800</td>
<td>3.33</td>
<td>172.82</td>
<td>640</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>655.95</td>
<td>2740</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>3.84</td>
<td>655.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>4.50</td>
<td>597.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.03</td>
<td>7868.97</td>
</tr>
</tbody>
</table>

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
Variables
From_date, To_date, System_ID

The report contains this information:

**Horizontal axis**
Processor time, in seconds

**Vertical axis**
Department name

*Figure 109. Sample Report 3*
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

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF_000</td>
<td>DRLRS000</td>
<td>IPL</td>
</tr>
<tr>
<td>SMF_002</td>
<td>DRLRS002</td>
<td>Dump header</td>
</tr>
<tr>
<td>SMF_003</td>
<td>DRLRS003</td>
<td>Dump trailer</td>
</tr>
<tr>
<td>SMF_004</td>
<td>DRLRS004</td>
<td>Step termination</td>
</tr>
<tr>
<td>SMF_005</td>
<td>DRLRS005</td>
<td>Job termination</td>
</tr>
<tr>
<td>SMF_006</td>
<td>DRLRS006</td>
<td>JES2/JES3/PSF/External writer</td>
</tr>
<tr>
<td>SMF_007</td>
<td>DRLRS007</td>
<td>Data lost</td>
</tr>
<tr>
<td>SMF_008</td>
<td>DRLRS008</td>
<td>I/O configuration</td>
</tr>
<tr>
<td>SMF_009</td>
<td>DRLRS009</td>
<td>VARY device ONLINE</td>
</tr>
<tr>
<td>SMF_010</td>
<td>DRLRS010</td>
<td>Allocation recovery</td>
</tr>
<tr>
<td>SMF_011</td>
<td>DRLRS011</td>
<td>VARY device OFFLINE</td>
</tr>
<tr>
<td>SMF_014</td>
<td>DRLRS014</td>
<td>INPUT or RDBACK data set activity</td>
</tr>
<tr>
<td>SMF_015</td>
<td>DRLRS015</td>
<td>OUTPUT, UPDAT, INOUT, or OUTIN data set activity</td>
</tr>
<tr>
<td>SMF_016</td>
<td>DRLRS016</td>
<td>DFSORT statistics</td>
</tr>
<tr>
<td>SMF_017</td>
<td>DRLRS017</td>
<td>Scratch data set status</td>
</tr>
<tr>
<td>SMF_018</td>
<td>DRLRS018</td>
<td>Rename data set status</td>
</tr>
<tr>
<td>SMF_019</td>
<td>DRLRS019</td>
<td>Direct access volume</td>
</tr>
<tr>
<td>SMF_020</td>
<td>DRLRS020</td>
<td>Job initiation</td>
</tr>
<tr>
<td>SMF_021</td>
<td>DRLRS021</td>
<td>Error statistics by volume</td>
</tr>
<tr>
<td>SMF_022</td>
<td>DRLRS022</td>
<td>Configuration</td>
</tr>
<tr>
<td>SMF_023</td>
<td>DRLRS023</td>
<td>SMF status</td>
</tr>
<tr>
<td>SMF_024</td>
<td>DRLRS024</td>
<td>JES2 spool offload</td>
</tr>
<tr>
<td>SMF_025</td>
<td>DRLRS025</td>
<td>JES3 device allocation</td>
</tr>
<tr>
<td>SMF_026</td>
<td>DRLRS026</td>
<td>JES2/JES3 job purge</td>
</tr>
<tr>
<td>SMF_028</td>
<td>DRLRS028</td>
<td>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.</td>
</tr>
<tr>
<td>SMF_028_NTRI</td>
<td>DRLRSNTR</td>
<td>NPM NTRI statistics</td>
</tr>
<tr>
<td>SMF_028_TRANSIT</td>
<td>DRLRSNTT</td>
<td>NPM transit time statistics</td>
</tr>
</tbody>
</table>
### SMF records

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF_028_TRANS_SUM</td>
<td>DRLRSNT1</td>
<td>NPM Transit Time summary statistics</td>
</tr>
<tr>
<td>SMF_028_X25</td>
<td>DRLRSX25</td>
<td>NPM X25 statistics</td>
</tr>
<tr>
<td>SMF_028_PU</td>
<td>DRLRSNPU</td>
<td>NPM PU statistics</td>
</tr>
<tr>
<td>SMF_028_NPM</td>
<td>DRLRSNPM</td>
<td>NPM internal statistics</td>
</tr>
<tr>
<td>SMF_028_LINE</td>
<td>DRLRSNL1</td>
<td>NPM line statistics</td>
</tr>
<tr>
<td>SMF_028_NEO</td>
<td>DRLRSNEO</td>
<td>NPM NEO statistics</td>
</tr>
<tr>
<td>SMF_028_NCP</td>
<td>DRLRSNCP</td>
<td>NPM NCP statistics</td>
</tr>
<tr>
<td>SMF_028_LAN</td>
<td>DRLRSLAN</td>
<td>NPM LAN statistics</td>
</tr>
<tr>
<td>SMF_028_VTAM</td>
<td>DRLRSVTM</td>
<td>NPM VTAM statistics</td>
</tr>
<tr>
<td>SMF_030</td>
<td>DRLRS030</td>
<td>Common address space work</td>
</tr>
<tr>
<td>SMF_031</td>
<td>DRLRS031</td>
<td>TIOC initialization</td>
</tr>
<tr>
<td>SMF_032</td>
<td>DRLRS032</td>
<td>TSO user work accounting</td>
</tr>
<tr>
<td>SMF_033</td>
<td>DRLRS033</td>
<td>APPC/MVS TP accounting</td>
</tr>
<tr>
<td>SMF_034</td>
<td>DRLRS034</td>
<td>TS-step termination</td>
</tr>
<tr>
<td>SMF_035</td>
<td>DRLRS035</td>
<td>LOGOFF</td>
</tr>
<tr>
<td>SMF_036</td>
<td>DRLRS036</td>
<td>ICF catalog</td>
</tr>
<tr>
<td>SMF_037_HW</td>
<td>DRLRS037</td>
<td>NetView Hardware Monitor</td>
</tr>
<tr>
<td>SMF_037_VPD</td>
<td>DRLRSVPD</td>
<td>Network configuration (VPD)</td>
</tr>
<tr>
<td>SMF_039_1_TO_7</td>
<td>DRLRS039</td>
<td>NetView Session Monitor, SMF 39, subtypes 1 to 7</td>
</tr>
<tr>
<td>SMF_039_8</td>
<td>DRLRS039</td>
<td>NetView Session Monitor, SMF 39, subtype 8</td>
</tr>
<tr>
<td>SMF_040</td>
<td>DRLRS040</td>
<td>Dynamic DD</td>
</tr>
<tr>
<td>SMF_041</td>
<td>DRLRS041</td>
<td>Data-in-virtual Access/Unaccess</td>
</tr>
<tr>
<td>SMF_042_1</td>
<td>DRLRS042</td>
<td>BMF performance statistics</td>
</tr>
<tr>
<td>SMF_042_2</td>
<td>DRLRS042</td>
<td>DFP cache control unit statistics</td>
</tr>
<tr>
<td>SMF_042_3</td>
<td>DRLRS042</td>
<td>DFP SMS configuration statistics</td>
</tr>
<tr>
<td>SMF_042_5</td>
<td>DRLRSX42</td>
<td>DFSMS storage class statistics</td>
</tr>
<tr>
<td>SMF_042_6</td>
<td>DRLRSX42</td>
<td>DFSMS Data Set statistics</td>
</tr>
<tr>
<td>SMF_042_14</td>
<td>DRLRADSM</td>
<td>ADSTAR Distributed Storage Manager (ADSM) server stats</td>
</tr>
<tr>
<td>SMF_042_11</td>
<td>DRLRSX42</td>
<td>DFP Extended Remote Copy (XRC) session statistics</td>
</tr>
<tr>
<td>SMF_043_2</td>
<td>DRLRS043</td>
<td>JES2 start</td>
</tr>
<tr>
<td>SMF_043_5</td>
<td>DRLRS043</td>
<td>JES3 start</td>
</tr>
<tr>
<td>SMF_045_2</td>
<td>DRLRS045</td>
<td>JES2 withdrawal</td>
</tr>
<tr>
<td>SMF_045_5</td>
<td>DRLRS045</td>
<td>JES3 stop</td>
</tr>
<tr>
<td>SMF_047_2</td>
<td>DRLRS047</td>
<td>JES2 SIGNON/start line (BSC only)</td>
</tr>
<tr>
<td>SMF_047_5</td>
<td>DRLRS047</td>
<td>JES3 SIGNON/start line/LOGON</td>
</tr>
<tr>
<td>SMF_048_2</td>
<td>DRLRS048</td>
<td>JES2 SIGNOFF/stop line (BSC only)</td>
</tr>
<tr>
<td>SMF_048_5</td>
<td>DRLRS048</td>
<td>JES3 SIGNOFF/stop line/LOGOFF</td>
</tr>
<tr>
<td>SMF_049_2</td>
<td>DRLRS049</td>
<td>JES2 integrity (BSC only)</td>
</tr>
<tr>
<td>SMF_049_5</td>
<td>DRLRS049</td>
<td>JES3 integrity</td>
</tr>
<tr>
<td>SMF_050</td>
<td>DRLRS050</td>
<td>ACF/VTAM* tuning statistics</td>
</tr>
<tr>
<td>SMF_052</td>
<td>DRLRS052</td>
<td>JES2 LOGON/start line (SNA only)</td>
</tr>
<tr>
<td>Record name</td>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SMF_053</td>
<td>DRLRS053</td>
<td>JES2 LOGOFF/start line (SNA only)</td>
</tr>
<tr>
<td>SMF_054</td>
<td>DRLRS054</td>
<td>JES2 integrity (SNA only)</td>
</tr>
<tr>
<td>SMF_055</td>
<td>DRLRS055</td>
<td>JES2 network SIGNON</td>
</tr>
<tr>
<td>SMF_056</td>
<td>DRLRS056</td>
<td>JES2 network integrity</td>
</tr>
<tr>
<td>SMF_057_2</td>
<td>DRLRS057</td>
<td>JES2 network SYSOUT transmission</td>
</tr>
<tr>
<td>SMF_057_5</td>
<td>DRLRS057</td>
<td>JES3 networking transmission</td>
</tr>
<tr>
<td>SMF_058</td>
<td>DRLRS058</td>
<td>JES2 network SIGNOFF</td>
</tr>
<tr>
<td>SMF_059</td>
<td>DRLRS059</td>
<td>MVS/BDT file-to-file transmission</td>
</tr>
<tr>
<td>SMF_060</td>
<td>DRLRS060</td>
<td>VSAM volume data set updated</td>
</tr>
<tr>
<td>SMF_061</td>
<td>DRLRS061</td>
<td>ICF define activity</td>
</tr>
<tr>
<td>SMF_062</td>
<td>DRLRS062</td>
<td>VSAM component or cluster opened</td>
</tr>
<tr>
<td>SMF_063</td>
<td>DRLRS063</td>
<td>VSAM entry defined</td>
</tr>
<tr>
<td>SMF_064</td>
<td>DRLRS064</td>
<td>VSAM component or cluster status</td>
</tr>
<tr>
<td>SMF_065</td>
<td>DRLRS065</td>
<td>ICF delete activity</td>
</tr>
<tr>
<td>SMF_066</td>
<td>DRLRS066</td>
<td>ICF alter activity</td>
</tr>
<tr>
<td>SMF_067</td>
<td>DRLRS067</td>
<td>VSAM entry delete</td>
</tr>
<tr>
<td>SMF_068</td>
<td>DRLRS068</td>
<td>VSAM entry renamed</td>
</tr>
<tr>
<td>SMF_069</td>
<td>DRLRS069</td>
<td>VSAM data space defined, extended, or deleted</td>
</tr>
<tr>
<td>SMF_070</td>
<td>DRLRS070</td>
<td>RMF™ CPU activity</td>
</tr>
<tr>
<td>SMF_071</td>
<td>DRLRS071</td>
<td>RMF paging activity</td>
</tr>
<tr>
<td>SMF_072_1</td>
<td>DRLRS072</td>
<td>RMF workload activity</td>
</tr>
<tr>
<td>SMF_072_2</td>
<td>DRLRSX72</td>
<td>RMF storage data</td>
</tr>
<tr>
<td>SMF_072_3</td>
<td>DRLRS072</td>
<td>RMF goal mode workload activity</td>
</tr>
<tr>
<td>SMF_072_4</td>
<td>DRLRSX72</td>
<td>RMF goal mode delay and storage frame data</td>
</tr>
<tr>
<td>SMF_073</td>
<td>DRLRS073</td>
<td>RMF channel path activity</td>
</tr>
<tr>
<td>SMF_074_1</td>
<td>DRLRS074</td>
<td>RMF device activity</td>
</tr>
<tr>
<td>SMF_074_2</td>
<td>DRLRS074</td>
<td>RMF XCF activity</td>
</tr>
<tr>
<td>SMF_074_3</td>
<td>DRLRSX74</td>
<td>RMF Device OMVS activity</td>
</tr>
<tr>
<td>SMF_074_4</td>
<td>DRLRSX74</td>
<td>RMF XES/CF activity</td>
</tr>
<tr>
<td>SMF_074_6</td>
<td>DRLRX74</td>
<td>File system statistics</td>
</tr>
<tr>
<td>SMF_075</td>
<td>DRLRS075</td>
<td>RMF page/swap data set activity</td>
</tr>
<tr>
<td>SMF_076</td>
<td>DRLRS076</td>
<td>RMF trace activity</td>
</tr>
<tr>
<td>SMF_077</td>
<td>DRLRS077</td>
<td>RMF enqueue activity</td>
</tr>
<tr>
<td>SMF_078_1</td>
<td>DRLRS078</td>
<td>RMF I/O queueing activity for the 308x, 908x, and 4381 processors</td>
</tr>
<tr>
<td>SMF_078_2</td>
<td>DRLRS078</td>
<td>RMF virtual storage activity</td>
</tr>
<tr>
<td>SMF_078_3</td>
<td>DRLRS078</td>
<td>RMF I/O queueing activity for the 3090, 9021, 9121, and 9221 processors</td>
</tr>
<tr>
<td>SMF_079</td>
<td>DRLRS079</td>
<td>RMF Monitor II activity</td>
</tr>
<tr>
<td>SMF_080</td>
<td>DRLRS080</td>
<td>RACF processing</td>
</tr>
<tr>
<td>SMF_081</td>
<td>DRLRS081</td>
<td>RACF initialization</td>
</tr>
<tr>
<td>SMF_082_1</td>
<td>DRLRS082</td>
<td>PCF record</td>
</tr>
</tbody>
</table>
### SMF records

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF_082_2</td>
<td>DRLRS082</td>
<td>CUSP record</td>
</tr>
<tr>
<td>SMF_083</td>
<td>DRLRS083</td>
<td>RACF audit record for data sets</td>
</tr>
<tr>
<td>SMF_084_1</td>
<td>DRLRS084</td>
<td>JMF - FCT analysis</td>
</tr>
<tr>
<td>SMF_084_2</td>
<td>DRLRS084</td>
<td>JMF - FCT summary and highlights</td>
</tr>
<tr>
<td>SMF_084_3</td>
<td>DRLRS084</td>
<td>JMF - spool data management</td>
</tr>
<tr>
<td>SMF_084_4</td>
<td>DRLRS084</td>
<td>JMF - resqueue cellpool, JCT and control block utilization</td>
</tr>
<tr>
<td>SMF_084_5</td>
<td>DRLRS084</td>
<td>JMF - job analysis</td>
</tr>
<tr>
<td>SMF_084_6</td>
<td>DRLRS084</td>
<td>JMF - JES3 hot spot analysis</td>
</tr>
<tr>
<td>SMF_084_7</td>
<td>DRLRS084</td>
<td>JMF - JES internal reader DSP analysis</td>
</tr>
<tr>
<td>SMF_084_8</td>
<td>DRLRS084</td>
<td>JMF - JES3 SSI response time analysis</td>
</tr>
<tr>
<td>SMF_084_9</td>
<td>DRLRS084</td>
<td>JMF - JES3 SSI destination queue analysis</td>
</tr>
<tr>
<td>SMF_085</td>
<td>DRLRS085</td>
<td>OAM record</td>
</tr>
<tr>
<td>SMF_088</td>
<td>DRLRS088</td>
<td>System logger</td>
</tr>
<tr>
<td>SMF_089</td>
<td>DRLRS089</td>
<td>Product Usage Data</td>
</tr>
<tr>
<td>SMF_090</td>
<td>DRLRS090</td>
<td>System status</td>
</tr>
<tr>
<td>SMF_092</td>
<td>DRLRS092</td>
<td>z/OS UNIX activity</td>
</tr>
<tr>
<td>SMF_094</td>
<td>DRLRS094</td>
<td>3494, 3495 Tape library data server statistics</td>
</tr>
<tr>
<td>SMF_099</td>
<td>DRLRS099</td>
<td>SMS System Resource Manager decisions</td>
</tr>
<tr>
<td>SMF_100_0</td>
<td>DRLRS100</td>
<td>DB2 statistics, system services</td>
</tr>
<tr>
<td>SMF_100_1</td>
<td>DRLRS100</td>
<td>DB2 statistics, database services</td>
</tr>
<tr>
<td>SMF_100_2</td>
<td>DRLRS100</td>
<td>DB2 statistics, dynamic ZPARMs</td>
</tr>
<tr>
<td>SMF_100_3</td>
<td>DRLRS100</td>
<td>DB2 statistics, Buffer, Manager Group Buffer Pool</td>
</tr>
<tr>
<td>SMF_101</td>
<td>DRLRS101</td>
<td>DB2 accounting</td>
</tr>
<tr>
<td>SMF_101_1</td>
<td>DRLRS101</td>
<td>DB2 accounting, Packages extension</td>
</tr>
<tr>
<td>SMF_102</td>
<td>DRLRS102</td>
<td>DB2 system initialization parameters</td>
</tr>
<tr>
<td>SMF_110_0</td>
<td>DRLRS110</td>
<td>CICS/ESA journaling record</td>
</tr>
<tr>
<td>SMF_110_0_V2</td>
<td>DRLRS110</td>
<td>CICS/MVS monitoring record</td>
</tr>
<tr>
<td>SMF_110_1</td>
<td>DRLRS110</td>
<td>CICS/ESA monitoring record</td>
</tr>
<tr>
<td>SMF_110_1_1</td>
<td>DRLRS110</td>
<td>CICS/TS &lt;3.2 record</td>
</tr>
<tr>
<td>SMF_110_1_5</td>
<td>DRLRS110T</td>
<td>CICS transaction resource - expanded</td>
</tr>
<tr>
<td>SMF_110_2</td>
<td>DRLRS1102</td>
<td>CICS/ESA and CICS/TS statistics record</td>
</tr>
<tr>
<td>SMF_110_3</td>
<td>DRLRS1103</td>
<td>CICS/TS statistics record</td>
</tr>
<tr>
<td>SMF_110_4</td>
<td>DRLRS1103</td>
<td>CICS/TS CF statistics record</td>
</tr>
<tr>
<td>SMF_110_5</td>
<td>DRLRS1103</td>
<td>CICS/TS NC statistics record</td>
</tr>
<tr>
<td>SMF_110_1_C</td>
<td>DRLRS110</td>
<td>CICS/TS 3.2+ - may be compressed</td>
</tr>
<tr>
<td>SMF_110_1_CO</td>
<td>DRLRS110</td>
<td>CICS/TS 3.2+ - expanded</td>
</tr>
<tr>
<td>SMF_110_E</td>
<td>DRLRS110</td>
<td>CICS/ESA exception record - expanded</td>
</tr>
<tr>
<td>SMF_112_203_C</td>
<td>DRLRS112</td>
<td>OMEGAMON® XE for CICS file and database usage – compressed</td>
</tr>
<tr>
<td>SMF_112_203</td>
<td>DRLRS112</td>
<td>OMEGAMON XE for CICS file and database usage – expanded</td>
</tr>
<tr>
<td>SMF_114_1</td>
<td>DRLRS114</td>
<td>System Automation Tracking</td>
</tr>
<tr>
<td>SMF_115</td>
<td>DRLRS115</td>
<td>WebSphere MQ for z/OS statistics</td>
</tr>
<tr>
<td>Record name</td>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SMF_116</td>
<td>DRLRS116</td>
<td>WebSphere MQ for z/OS statistics</td>
</tr>
<tr>
<td>SMF_117</td>
<td>DRLRS117</td>
<td>Websphere Message Broker</td>
</tr>
<tr>
<td>SMF_118_1</td>
<td>DRLRS118</td>
<td>TCP/IP API calls record</td>
</tr>
<tr>
<td>SMF_118_3</td>
<td>DRLRS118</td>
<td>TCP/IP FTP client calls record</td>
</tr>
<tr>
<td>SMF_118_4</td>
<td>DRLRS118</td>
<td>TCP/IP TELNET client calls record</td>
</tr>
<tr>
<td>SMF_118_20</td>
<td>DRLRS118</td>
<td>TCP/IP TELNET server record</td>
</tr>
<tr>
<td>SMF_118_70</td>
<td>DRLRS118</td>
<td>TCP/IP FTP server record</td>
</tr>
<tr>
<td>SMF_119_1</td>
<td>DRLRS119</td>
<td>TCP connection initiation</td>
</tr>
<tr>
<td>SMF_119_2</td>
<td>DRLRS119</td>
<td>TCP connection termination</td>
</tr>
<tr>
<td>SMF_119_3</td>
<td>DRLRS119</td>
<td>FTP client transfer completion</td>
</tr>
<tr>
<td>SMF_119_4</td>
<td>DRLRS119</td>
<td>TCP/IP Profile Information record</td>
</tr>
<tr>
<td>SMF_119_5</td>
<td>DRLRS119</td>
<td>TCP/IP statistics</td>
</tr>
<tr>
<td>SMF_119_6</td>
<td>DRLRS119</td>
<td>Interface statistics</td>
</tr>
<tr>
<td>SMF_119_7</td>
<td>DRLRS119</td>
<td>Server port statistics</td>
</tr>
<tr>
<td>SMF_119_8</td>
<td>DRLRS119</td>
<td>TCP/IP stack start/stop</td>
</tr>
<tr>
<td>SMF_119_10</td>
<td>DRLRS119</td>
<td>UDP socket close</td>
</tr>
<tr>
<td>SMF_119_20</td>
<td>DRLRS119</td>
<td>TN3270 server SNA session initiation</td>
</tr>
<tr>
<td>SMF_119_21</td>
<td>DRLRS119</td>
<td>TN3270 server SNA session termination</td>
</tr>
<tr>
<td>SMF_119_22</td>
<td>DRLRS119</td>
<td>TSO telnet client connection initiation</td>
</tr>
<tr>
<td>SMF_119_23</td>
<td>DRLRS119</td>
<td>TSO telnet client connection termination</td>
</tr>
<tr>
<td>SMF_119_70</td>
<td>DRLRS119</td>
<td>FTP server transfer completion</td>
</tr>
<tr>
<td>SMF_119_72</td>
<td>DRLRS119</td>
<td>FTP server logon failure</td>
</tr>
<tr>
<td>SMF_119_73</td>
<td>DRLRS119</td>
<td>IPSec IKE Tunnel Activation/Refresh record</td>
</tr>
<tr>
<td>SMF_119_74</td>
<td>DRLRS119</td>
<td>IPSec IKE Tunnel Deactivation/Expire record</td>
</tr>
<tr>
<td>SMF_119_75_80</td>
<td>DRLRS119</td>
<td>IPSec Dynamic Tunnel Activation/Refresh</td>
</tr>
<tr>
<td>SMF_119_75_80</td>
<td>DRLRS119</td>
<td>IPSec Dynamic Tunnel Deactivation record</td>
</tr>
<tr>
<td>SMF_119_75_80</td>
<td>DRLRS119</td>
<td>IPSec Dynamic Tunnel Added record</td>
</tr>
<tr>
<td>SMF_119_75_80</td>
<td>DRLRS119</td>
<td>IPSec Dynamic Tunnel Removed record</td>
</tr>
<tr>
<td>SMF_119_75_80</td>
<td>DRLRS119</td>
<td>IPSec Manual Tunnel Activation record</td>
</tr>
<tr>
<td>SMF_119_75_80</td>
<td>DRLRS119</td>
<td>IPSec Manual Tunnel Deactivation record</td>
</tr>
<tr>
<td>SMF_120_1</td>
<td>DRLRS121</td>
<td>Server activity record</td>
</tr>
<tr>
<td>SMF_120_2</td>
<td>DRLRS122</td>
<td>WebSphere Application Server container activity record</td>
</tr>
<tr>
<td>SMF_120_3</td>
<td>DRLRS123</td>
<td>Server interval record</td>
</tr>
<tr>
<td>SMF_120_4</td>
<td>DRLRS124</td>
<td>WebSphere Application Server container interval record</td>
</tr>
<tr>
<td>SMF_120_5</td>
<td>DRLRSJWA</td>
<td>J2EE container activity record</td>
</tr>
<tr>
<td>SMF_120_6</td>
<td>DRLRSJWI</td>
<td>J2EE container interval record</td>
</tr>
<tr>
<td>SMF_120_7</td>
<td>DRLRSJWA</td>
<td>Web container activity record</td>
</tr>
<tr>
<td>SMF_120_8</td>
<td>DRLRSJWI</td>
<td>Web container interval record</td>
</tr>
<tr>
<td>SMF_120_9</td>
<td>DRLRS129</td>
<td>Request Activity record</td>
</tr>
<tr>
<td>SMF_120_10</td>
<td>DRLRS12A</td>
<td>Outbound Request record</td>
</tr>
</tbody>
</table>
SMF records

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF_120_11</td>
<td>DRLRSRWL</td>
<td>Liberty request record</td>
<td></td>
</tr>
<tr>
<td>SMF_120_12</td>
<td>DRLRSJWL</td>
<td>Liberty Java batch job record</td>
<td></td>
</tr>
<tr>
<td>SMF_123</td>
<td>DRLRS123</td>
<td>SMF HPQS statistics</td>
<td></td>
</tr>
<tr>
<td>SMF_194</td>
<td>DRLRS194</td>
<td>TS7700 Virtualization Engine statistics record</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_01</td>
<td>DRLRIXFP</td>
<td>IXFP subsystem performance</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_02</td>
<td>DRLRIXFP</td>
<td>IXFP channel interface statistics</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_03</td>
<td>DRLRIXFP</td>
<td>IXFP functional device performance</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_04</td>
<td>DRLRIXFP</td>
<td>IXFP device module performance</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_05</td>
<td>DRLRIXFP</td>
<td>IXFP deleted data space release</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_06</td>
<td>DRLRIXFP</td>
<td>IXFP snapshot event data</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_07</td>
<td>DRLRIXFP</td>
<td>IXFP space utilization record</td>
<td></td>
</tr>
<tr>
<td>SMF_IXFP_08</td>
<td>DRLRIXFP</td>
<td>IXFP snapshot extended event data record</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF_CACHE_03</td>
<td>DRLRS245</td>
<td>Cache RMF Reporter, 3990 model 03 (245)</td>
<td></td>
</tr>
<tr>
<td>SMF_CACHE_06</td>
<td>DRLRS245</td>
<td>Cache RMF Reporter, 3990 model 06 (245)</td>
<td></td>
</tr>
<tr>
<td>SMF_CACHE_13</td>
<td>DRLRS245</td>
<td>Cache RMF Reporter, 3880 model 13 (245)</td>
<td></td>
</tr>
<tr>
<td>SMF_CACHE_23</td>
<td>DRLRS245</td>
<td>Cache RMF Reporter, 3880 model 23 (245)</td>
<td></td>
</tr>
<tr>
<td>SMF_FTP</td>
<td>DRLRSFTP</td>
<td>NetView File Transfer Program (FTP) log record (252)</td>
<td></td>
</tr>
</tbody>
</table>

DFSMS/RMM records

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DFRMM_VOLUME</td>
<td>DRLRRMMV</td>
<td>Extract file volume record</td>
<td></td>
</tr>
<tr>
<td>DFRMM_RACK</td>
<td>DRLRRMMR</td>
<td>Extract file rack number record</td>
<td></td>
</tr>
<tr>
<td>DFRMM_SLBIN</td>
<td>DRLRRMMS</td>
<td>Extract file storage location bin record</td>
<td></td>
</tr>
<tr>
<td>DFRMM_PRODUCT</td>
<td>DRLRRMMP</td>
<td>Extract file product record</td>
<td></td>
</tr>
<tr>
<td>DFRMM_VRS</td>
<td>DRLRRMK</td>
<td>Extract file VRS record</td>
<td></td>
</tr>
<tr>
<td>DFRMM_OWNER</td>
<td>DRLRMMO</td>
<td>Extract file owner record</td>
<td></td>
</tr>
<tr>
<td>DFRMM_DATASET</td>
<td>DRLRRMMD</td>
<td>Extract file dataset record</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_Vnn0_01</td>
<td>DRLRInnS</td>
<td>Message Queue record (message received from a CNT)</td>
</tr>
<tr>
<td>IMS_Vnn0_02</td>
<td>DRLRInnS</td>
<td>IMS command record</td>
</tr>
<tr>
<td>IMS_Vnn0_03</td>
<td>DRLRInnS</td>
<td>Message Queue record (message received from an SMB or IMS)</td>
</tr>
<tr>
<td>IMS_Vnn0_06</td>
<td>DRLRInnS</td>
<td>IMS event accounting record</td>
</tr>
<tr>
<td>IMS_Vnn0_07</td>
<td>DRLRInnS</td>
<td>Program termination accounting record</td>
</tr>
<tr>
<td>IMS_Vnn0_08</td>
<td>DRLRInnS</td>
<td>Program schedule record</td>
</tr>
<tr>
<td>IMS_Vnn0_10</td>
<td>DRLRInnS</td>
<td>Security violation record</td>
</tr>
<tr>
<td>IMS_Vnn0_11</td>
<td>DRLRInnS</td>
<td>Start of conversation record</td>
</tr>
<tr>
<td>IMS_Vnn0_12</td>
<td>DRLRInnS</td>
<td>End of conversation record</td>
</tr>
<tr>
<td>IMS_Vnn0_13</td>
<td>DRLRInnS</td>
<td>SPA insert record</td>
</tr>
<tr>
<td>IMS_Vnn0_16</td>
<td>DRLRInnS</td>
<td>Sign on/off record</td>
</tr>
<tr>
<td>IMS_Vnn0_18</td>
<td>DRLRInnS</td>
<td>Extended checkpoint record</td>
</tr>
<tr>
<td>IMS_Vnn0_20</td>
<td>DRLRInnS</td>
<td>Database open record</td>
</tr>
<tr>
<td>IMS_Vnn0_21</td>
<td>DRLRInnS</td>
<td>Database close record</td>
</tr>
<tr>
<td>IMS_Vnn0_24</td>
<td>DRLRInnS</td>
<td>Database error record</td>
</tr>
<tr>
<td>IMS_Vnn0_30</td>
<td>DRLRInnS</td>
<td>Message queue prefix changed record</td>
</tr>
<tr>
<td>IMS_Vnn0_31</td>
<td>DRLRInnS</td>
<td>Message queue GU record</td>
</tr>
<tr>
<td>IMS_Vnn0_32</td>
<td>DRLRInnS</td>
<td>Message queue reject record</td>
</tr>
<tr>
<td>IMS_Vnn0_33</td>
<td>DRLRInnS</td>
<td>Message queue DRRN free record</td>
</tr>
<tr>
<td>IMS_Vnn0_34</td>
<td>DRLRInnS</td>
<td>Message queue cancel record</td>
</tr>
<tr>
<td>IMS_Vnn0_35</td>
<td>DRLRInnS</td>
<td>Message queue enqueue record</td>
</tr>
<tr>
<td>IMS_Vnn0_36</td>
<td>DRLRInnS</td>
<td>Message queue dequeue record</td>
</tr>
<tr>
<td>IMS_Vnn0_37</td>
<td>DRLRInnS</td>
<td>Message queue syncpoint transfer record</td>
</tr>
<tr>
<td>IMS_Vnn0_38</td>
<td>DRLRInnS</td>
<td>Message queue syncpoint fail record</td>
</tr>
<tr>
<td>IMS_Vnn0_4C</td>
<td>DRLRInnS</td>
<td>Program/Database start/stop record</td>
</tr>
<tr>
<td>IMS_Vnn0_400D</td>
<td>DRLRInnS</td>
<td>Checkpoint CCB record</td>
</tr>
<tr>
<td>IMS_Vnn0_400E</td>
<td>DRLRInnS</td>
<td>Checkpoint SPA record</td>
</tr>
<tr>
<td>IMS_Vnn0_4001</td>
<td>DRLRInnS</td>
<td>Checkpoint begin</td>
</tr>
<tr>
<td>IMS_Vnn0_4002</td>
<td>DRLRInnS</td>
<td>Checkpoint message queue record</td>
</tr>
<tr>
<td>IMS_Vnn0_4003</td>
<td>DRLRInnS</td>
<td>Checkpoint CNT record</td>
</tr>
<tr>
<td>IMS_Vnn0_4004</td>
<td>DRLRInnS</td>
<td>Checkpoint SMB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4005</td>
<td>DRLRInnS</td>
<td>Checkpoint CTB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4006</td>
<td>DRLRInnS</td>
<td>Checkpoint DMB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4007</td>
<td>DRLRInnS</td>
<td>Checkpoint PSB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4008</td>
<td>DRLRInnS</td>
<td>Checkpoint CLB record</td>
</tr>
<tr>
<td>Record name</td>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>IMS_Vnn0_4014</td>
<td>DRLRInnS</td>
<td>Checkpoint SPA QB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4015</td>
<td>DRLRInnS</td>
<td>Checkpoint EQE record</td>
</tr>
<tr>
<td>IMS_Vnn0_4020</td>
<td>DRLRInnS</td>
<td>Checkpoint CIB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4021</td>
<td>DRLRInnS</td>
<td>Checkpoint VTCB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4070</td>
<td>DRLRInnS</td>
<td>Checkpoint MSDB begin</td>
</tr>
<tr>
<td>IMS_Vnn0_4071</td>
<td>DRLRInnS</td>
<td>Checkpoint MSDB ECNT record</td>
</tr>
<tr>
<td>IMS_Vnn0_4072</td>
<td>DRLRInnS</td>
<td>Checkpoint MSDB header</td>
</tr>
<tr>
<td>IMS_Vnn0_4073</td>
<td>DRLRInnS</td>
<td>Checkpoint MSDB pagefixed</td>
</tr>
<tr>
<td>IMS_Vnn0_4074</td>
<td>DRLRInnS</td>
<td>Checkpoint MSDB pageable</td>
</tr>
<tr>
<td>IMS_Vnn0_4079</td>
<td>DRLRInnS</td>
<td>Checkpoint MSDB end</td>
</tr>
<tr>
<td>IMS_Vnn0_4080</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path begin</td>
</tr>
<tr>
<td>IMS_Vnn0_4081</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path ECNT record</td>
</tr>
<tr>
<td>IMS_Vnn0_4082</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path EMHB record</td>
</tr>
<tr>
<td>IMS_Vnn0_4083</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path RCTE record</td>
</tr>
<tr>
<td>IMS_Vnn0_4084</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path DMCB/DMAC record</td>
</tr>
<tr>
<td>IMS_Vnn0_4085</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path MTO buffer record</td>
</tr>
<tr>
<td>IMS_Vnn0_4086</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path DMHR/DEDB buffer record</td>
</tr>
<tr>
<td>IMS_Vnn0_4087</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path ADSC record</td>
</tr>
<tr>
<td>IMS_Vnn0_4088</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path IEEQE record</td>
</tr>
<tr>
<td>IMS_Vnn0_4089</td>
<td>DRLRInnS</td>
<td>Checkpoint Fast Path end</td>
</tr>
<tr>
<td>IMS_Vnn0_4098</td>
<td>DRLRInnS</td>
<td>Checkpoint end blocks record</td>
</tr>
<tr>
<td>IMS_Vnn0_4099</td>
<td>DRLRInnS</td>
<td>Checkpoint end queues record</td>
</tr>
<tr>
<td>IMS_Vnn0_41</td>
<td>DRLRInnS</td>
<td>Checkpoint batch record</td>
</tr>
<tr>
<td>IMS_Vnn0_42</td>
<td>DRLRInnS</td>
<td>Log buffer control record</td>
</tr>
<tr>
<td>IMS_Vnn0_43</td>
<td>DRLRInnS</td>
<td>Log dataset control record</td>
</tr>
<tr>
<td>IMS_Vnn0_45FF</td>
<td>DRLRInnS</td>
<td>End of statistics</td>
</tr>
<tr>
<td>IMS_Vnn0_450A</td>
<td>DRLRInnS</td>
<td>Statistics latch record</td>
</tr>
<tr>
<td>IMS_Vnn0_450B</td>
<td>DRLRInnS</td>
<td>Statistics dispatch storage record</td>
</tr>
<tr>
<td>IMS_Vnn0_450C</td>
<td>DRLRInnS</td>
<td>Statistics DFSCBT00 storage record</td>
</tr>
<tr>
<td>IMS_Vnn0_450D</td>
<td>DRLRInnS</td>
<td>Statistics RecAny pool record</td>
</tr>
<tr>
<td>IMS_Vnn0_450E</td>
<td>DRLRInnS</td>
<td>Statistics fixed pools storage record</td>
</tr>
<tr>
<td>IMS_Vnn0_450F</td>
<td>DRLRInnS</td>
<td>Dispatcher statistics record</td>
</tr>
<tr>
<td>IMS_Vnn0_4502</td>
<td>DRLRInnS</td>
<td>Statistics queue pool record</td>
</tr>
<tr>
<td>IMS_Vnn0_4503</td>
<td>DRLRInnS</td>
<td>Statistics format buffer pool record</td>
</tr>
<tr>
<td>IMS_Vnn0_4504</td>
<td>DRLRInnS</td>
<td>Statistics database buffer pool</td>
</tr>
<tr>
<td>IMS_Vnn0_4505</td>
<td>DRLRInnS</td>
<td>Statistics main pools record</td>
</tr>
<tr>
<td>IMS_Vnn0_4506</td>
<td>DRLRInnS</td>
<td>Statistics scheduling stats record</td>
</tr>
<tr>
<td>IMS_Vnn0_4507</td>
<td>DRLRInnS</td>
<td>Statistics logger record</td>
</tr>
<tr>
<td>IMS_Vnn0_4508</td>
<td>DRLRInnS</td>
<td>Statistics VSAM subpool record</td>
</tr>
<tr>
<td>IMS_Vnn0_4509</td>
<td>DRLRInnS</td>
<td>Statistics program isolation record</td>
</tr>
<tr>
<td>IMS_Vnn0_47</td>
<td>DRLRInnS</td>
<td>Statistics active region record</td>
</tr>
<tr>
<td>Record name</td>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>IMS_Vnn0_48</td>
<td>DRLRInnS</td>
<td>OLDS padding record</td>
</tr>
<tr>
<td>IMS_Vnn0_5050</td>
<td>DRLRInnS</td>
<td>Full function database update undo/redo successful record</td>
</tr>
<tr>
<td>IMS_Vnn0_5051</td>
<td>DRLRInnS</td>
<td>Full function database update unsuccessful record</td>
</tr>
<tr>
<td>IMS_Vnn0_5052</td>
<td>DRLRInnS</td>
<td>Full function database update undo KSDS insert record</td>
</tr>
<tr>
<td>IMS_Vnn0_5501FE00</td>
<td>DRLRInnS</td>
<td>External sub-system DB2 snap in doubt record</td>
</tr>
<tr>
<td>IMS_Vnn0_56</td>
<td>DRLRInnS</td>
<td>External sub-system record</td>
</tr>
<tr>
<td>IMS_Vnn0_5901</td>
<td>DRLRInnS</td>
<td>EMH input record</td>
</tr>
<tr>
<td>IMS_Vnn0_5903</td>
<td>DRLRInnS</td>
<td>EMH output record</td>
</tr>
<tr>
<td>IMS_Vnn0_5920</td>
<td>DRLRInnS</td>
<td>Fast path MSDB change record</td>
</tr>
<tr>
<td>IMS_Vnn0_5921</td>
<td>DRLRInnS</td>
<td>Fast path DEDB area dataset open record</td>
</tr>
<tr>
<td>IMS_Vnn0_5922</td>
<td>DRLRInnS</td>
<td>Fast path DEDB area dataset close record</td>
</tr>
<tr>
<td>IMS_Vnn0_5923</td>
<td>DRLRInnS</td>
<td>Fast path DEDB area dataset status record</td>
</tr>
<tr>
<td>IMS_Vnn0_5924</td>
<td>DRLRInnS</td>
<td>Fast path DEDB area dataset EQE creation record</td>
</tr>
<tr>
<td>IMS_Vnn0_5936</td>
<td>DRLRInnS</td>
<td>EMH dequeue record</td>
</tr>
<tr>
<td>IMS_Vnn0_5937</td>
<td>DRLRInnS</td>
<td>EMH FP syncpoint record</td>
</tr>
<tr>
<td>IMS_Vnn0_5938</td>
<td>DRLRInnS</td>
<td>EMH FP syncpoint failure record</td>
</tr>
<tr>
<td>IMS_Vnn0_5950</td>
<td>DRLRInnS</td>
<td>Fast Path database update record</td>
</tr>
<tr>
<td>IMS_Vnn0_5953</td>
<td>DRLRInnS</td>
<td>Fast Path database update (utilities) record</td>
</tr>
<tr>
<td>IMS_Vnn0_5954</td>
<td>DRLRInnS</td>
<td>Fast Path database DEDB open record</td>
</tr>
<tr>
<td>IMS_Vnn0_5955</td>
<td>DRLRInnS</td>
<td>Fast Path sequential dependent syncpoint record</td>
</tr>
<tr>
<td>IMS_Vnn0_5957</td>
<td>DRLRInnS</td>
<td>Fast Path database DMAC record</td>
</tr>
<tr>
<td>IMS_Vnn0_5970</td>
<td>DRLRInnS</td>
<td>Fast Path hot standby MSDB relocation record</td>
</tr>
<tr>
<td>IMS_Vnn0_67</td>
<td>DRLRInnS</td>
<td>Communications trace, DMHR on I/O error and snap trace records</td>
</tr>
<tr>
<td>IMS_Vnn0_67FA</td>
<td>DRLRInnS</td>
<td>Trace table log record</td>
</tr>
<tr>
<td>IMS_Vnn0_7201</td>
<td>DRLRInnS</td>
<td>ETO user create record</td>
</tr>
<tr>
<td>IMS_Vnn0_7202</td>
<td>DRLRInnS</td>
<td>ETO user delete record</td>
</tr>
<tr>
<td>IMS_Vnn0_7203</td>
<td>DRLRInnS</td>
<td>ETO user modify record</td>
</tr>
<tr>
<td>IMS_Vnn0_7204</td>
<td>DRLRInnS</td>
<td>ETO lterm addition record</td>
</tr>
</tbody>
</table>

## 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*.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCOLLECT_A</td>
<td>DRLRDCOA</td>
<td>VSAM base cluster association name</td>
</tr>
<tr>
<td>DCOLLECT_AG</td>
<td>DRLRDCAG</td>
<td>Aggregate Group information</td>
</tr>
<tr>
<td>DCOLLECT_B</td>
<td>DRLRDCOB</td>
<td>Data set backup version information</td>
</tr>
<tr>
<td>DCOLLECT_BC</td>
<td>DRLRDCBC</td>
<td>Base Configuration information</td>
</tr>
<tr>
<td>DCOLLECT_C</td>
<td>DRLRDCOC</td>
<td>DASD capacity planning information</td>
</tr>
</tbody>
</table>
**DCOLLECT records**

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCOLLECT_D</td>
<td>DRLRDCOD</td>
<td>Active data set information</td>
</tr>
<tr>
<td>DCOLLECT_DC</td>
<td>DRLRDCC</td>
<td>Data Class construct information</td>
</tr>
<tr>
<td>DCOLLECT_DR</td>
<td>DRLRDCDR</td>
<td>Optical Drive information</td>
</tr>
<tr>
<td>DCOLLECT_LB</td>
<td>DRLRDLCB</td>
<td>Optical Library information</td>
</tr>
<tr>
<td>DCOLLECT_M</td>
<td>DRLRDCOM</td>
<td>Migration data set information</td>
</tr>
<tr>
<td>DCOLLECT_MC</td>
<td>DRLRDPMC</td>
<td>Management Class construct information</td>
</tr>
<tr>
<td>DCOLLECT_SC</td>
<td>DRLRDSCH</td>
<td>Storage Class construct information</td>
</tr>
<tr>
<td>DCOLLECT_SG</td>
<td>DRLRDSG</td>
<td>Storage Group construct information</td>
</tr>
<tr>
<td>DCOLLECT_T</td>
<td>DRLRDCT</td>
<td>Tape capacity planning information</td>
</tr>
<tr>
<td>DCOLLECT_V</td>
<td>DRLRDCOV</td>
<td>Volume information</td>
</tr>
<tr>
<td>DCOLLECT_VL</td>
<td>DRLRDCL</td>
<td>SMS Volume information</td>
</tr>
</tbody>
</table>

**EREP records**

For a description of these records, refer to the *Environmental Record Editing and Printing Program (EREP) User’s Guide and Reference.*

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EREP_30</td>
<td>DRLRE030</td>
<td>DASD long outboard record</td>
</tr>
<tr>
<td>EREP_36</td>
<td>DRLER036</td>
<td>VTAM long outboard record</td>
</tr>
<tr>
<td>EREP_50</td>
<td>DRLER050</td>
<td>IPL system initialization record</td>
</tr>
</tbody>
</table>

**Linux on zSeries records**

These records are produced by the zLinux programs on your zLinux nodes.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZLINUX_CPU</td>
<td>DRLRZPCP</td>
<td>zLinux CPU performance record</td>
</tr>
<tr>
<td>ZLINUX_DISK_FS</td>
<td>DRLRZPDI</td>
<td>zLinux disk space performance record</td>
</tr>
<tr>
<td>ZLINUX_DISKIO</td>
<td>DRLRZPIO</td>
<td>zLinux disk I/O performance record</td>
</tr>
<tr>
<td>ZLINUX_PAGING</td>
<td>DRLRZPPA</td>
<td>zLinux paging space performance record</td>
</tr>
<tr>
<td>ZLINUX_HARDCONF</td>
<td>DRLRZCNF</td>
<td>zLinux hardware configuration record</td>
</tr>
<tr>
<td>ZLINUX_SOFTCONF</td>
<td>DRLRZCNF</td>
<td>zLinux software configuration record</td>
</tr>
<tr>
<td>ZLINUX_USR_CMD</td>
<td>DRLRZACO</td>
<td>zLinux process/command accounting record</td>
</tr>
<tr>
<td>ZLINUX_WTMP_INFO</td>
<td>DRLRZMTP</td>
<td>zLinux connect accounting record</td>
</tr>
<tr>
<td>ZLINUX_REC_PI</td>
<td>DRLRLNX1</td>
<td>PI log record reformatted to fixed layout</td>
</tr>
<tr>
<td>ZLINUX_REC_DF</td>
<td>DRLRLNX1</td>
<td>DF log record reformatted to fixed layout</td>
</tr>
<tr>
<td>ZLINUX_REC_WW</td>
<td>DRLRLNX1</td>
<td>WW log record reformatted to fixed layout</td>
</tr>
<tr>
<td>ZLINUX_REC_TO</td>
<td>DRLRLNX1</td>
<td>TO log record reformatted to fixed layout</td>
</tr>
</tbody>
</table>
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*.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACF_100</td>
<td>DRLRR100</td>
<td>Group basic data</td>
</tr>
<tr>
<td>RACF_200</td>
<td>DRLRR200</td>
<td>User basic data</td>
</tr>
<tr>
<td>RACF_205</td>
<td>DRLRR205</td>
<td>User connect data</td>
</tr>
<tr>
<td>RACF_207</td>
<td>DRLRR207</td>
<td></td>
</tr>
<tr>
<td>RACF_400</td>
<td>DRLRR400</td>
<td>Data set basic data</td>
</tr>
<tr>
<td>RACF_402</td>
<td>DRLRR402</td>
<td>Data set conditional access</td>
</tr>
<tr>
<td>RACF_404</td>
<td>DRLRR404</td>
<td>Data set access</td>
</tr>
<tr>
<td>RACF_500</td>
<td>DRLRR500</td>
<td>General resource basic data</td>
</tr>
<tr>
<td>RACF_505</td>
<td>DRLRR505</td>
<td>General resource access</td>
</tr>
<tr>
<td>RACF_507</td>
<td>DRLRR507</td>
<td>General resource conditional access</td>
</tr>
</tbody>
</table>

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*.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPC_03_P</td>
<td>DRLROP03</td>
<td>OPC current plan operation</td>
</tr>
<tr>
<td>OPC_03_C</td>
<td>DRLROP03</td>
<td>OPC current plan occurrence</td>
</tr>
<tr>
<td>OPC_03_3</td>
<td>DRLROP03</td>
<td>OPC current plan system automation</td>
</tr>
<tr>
<td>OPC_04</td>
<td>DRLROP04</td>
<td>OPC current plan job name table</td>
</tr>
<tr>
<td>OPC_23</td>
<td>DRLROP23</td>
<td>OPC operation event</td>
</tr>
<tr>
<td>OPC_24</td>
<td>DRLROP24</td>
<td>OPC MCP event</td>
</tr>
<tr>
<td>OPC_27</td>
<td>DRLROP27</td>
<td>OPC missed feedback</td>
</tr>
<tr>
<td>OPC_29</td>
<td>DRLROP29</td>
<td>OPC auto tracked event</td>
</tr>
</tbody>
</table>

VM accounting records

For a description of these records, refer to *z/VM: CP Planning and Administration*.

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMACCT_01</td>
<td>DRLRVA01</td>
<td>Virtual machine resource use</td>
</tr>
<tr>
<td>VMACCT_02</td>
<td>DRLRVA02</td>
<td>Dedicated devices</td>
</tr>
<tr>
<td>VMACCT_03</td>
<td>DRLRVA03</td>
<td>Temporary disk space</td>
</tr>
<tr>
<td>VMACCT_04</td>
<td>DRLRVA04</td>
<td>LOGON or AUTOLOG with invalid password</td>
</tr>
<tr>
<td>VMACCT_05</td>
<td>DRLRVA05</td>
<td>Successful LINK to protected minidisk</td>
</tr>
<tr>
<td>VMACCT_06</td>
<td>DRLRVA06</td>
<td>LINK with invalid password</td>
</tr>
</tbody>
</table>
### VM accounting records

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMACCT_07</td>
<td>DRLRVA07</td>
<td>Log off from VSCS-controlled device</td>
</tr>
<tr>
<td>VMACCT_08</td>
<td>DRLRVA08</td>
<td>Disconnect or log off</td>
</tr>
</tbody>
</table>

### VMPRF records

For a description of these records, refer to the *VMPRF User’s Guide and Reference.*

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPRF_01</td>
<td>DRLRVM01</td>
<td>VMPRF system data</td>
</tr>
<tr>
<td>VMPRF_02</td>
<td>DRLRVM02</td>
<td>VMPRF processor data</td>
</tr>
<tr>
<td>VMPRF_11</td>
<td>DRLRVM11</td>
<td>VMPRF configuration data</td>
</tr>
<tr>
<td>VMPRF_41</td>
<td>DRLRVM41</td>
<td>VMPRF user data</td>
</tr>
<tr>
<td>VMPRF_61</td>
<td>DRLRVM61</td>
<td>VMPRF DASD data</td>
</tr>
</tbody>
</table>

### z/VM Performance Toolkit records

For a description of these records, refer to the *z/VM Performance Toolkit manual.*

<table>
<thead>
<tr>
<th>Record name</th>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPERFT_00</td>
<td>DRLRPT00</td>
<td>System configuration data</td>
</tr>
<tr>
<td>VMPERFT_01</td>
<td>DRLRPT01</td>
<td>General system load data</td>
</tr>
<tr>
<td>VMPERFT_02</td>
<td>DRLRPT02</td>
<td>Processor load data</td>
</tr>
<tr>
<td>VMPERFT_03</td>
<td>DRLRPT03</td>
<td>Logical processor load data (LPAR only)</td>
</tr>
<tr>
<td>VMPERFT_04</td>
<td>DRLRPT04</td>
<td>Minidisk cache data</td>
</tr>
<tr>
<td>VMPERFT_05</td>
<td>DRLRPT05</td>
<td>CP services activity data</td>
</tr>
<tr>
<td>VMPERFT_06</td>
<td>DRLRPT06</td>
<td>Channel busy (HF sampling)</td>
</tr>
<tr>
<td>VMPERFT_07</td>
<td>DRLRPT07</td>
<td>Channel measurement facility data</td>
</tr>
<tr>
<td>VMPERFT_08</td>
<td>DRLRPT08</td>
<td>Extended channel measurement facility data</td>
</tr>
<tr>
<td>VMPERFT_3A</td>
<td>DRLRPT3A</td>
<td>Overall user transaction data</td>
</tr>
<tr>
<td>VMPERFT_3C</td>
<td>DRLRPT3C</td>
<td>Shared segment data</td>
</tr>
<tr>
<td>VMPERFT_3E</td>
<td>DRLRPT3E</td>
<td>Shared data spaces</td>
</tr>
<tr>
<td>VMPERFT_41</td>
<td>DRLRPT41</td>
<td>User resource usage and wait states</td>
</tr>
<tr>
<td>VMPERFT_42</td>
<td>DRLRPT42</td>
<td>User class resource usage and wait states (same layout as FC41)</td>
</tr>
<tr>
<td>VMPERFT_43</td>
<td>DRLRPT43</td>
<td>System totals for user resource usage and wait states (same layout as FC41)</td>
</tr>
<tr>
<td>VMPERFT_44</td>
<td>DRLRPT44</td>
<td>User transactions and response time</td>
</tr>
<tr>
<td>VMPERFT_45</td>
<td>DRLRPT45</td>
<td>User class transactions and response time data (same layout as FC44)</td>
</tr>
<tr>
<td>VMPERFT_46</td>
<td>DRLRPT46</td>
<td>System totals for user transactions and response time data</td>
</tr>
<tr>
<td>VMPERFT_51</td>
<td>DRLRPT51</td>
<td>I/O processor activity data</td>
</tr>
<tr>
<td>VMPERFT_55</td>
<td>DRLRPT55</td>
<td>Virtual switch records</td>
</tr>
<tr>
<td>VMPERFT_61</td>
<td>DRLRPT61</td>
<td>General DASD data</td>
</tr>
<tr>
<td>VMPERFT_65</td>
<td>DRLRPT65</td>
<td>DASD cache data</td>
</tr>
<tr>
<td>VMPERFT_68</td>
<td>DRLRPT68</td>
<td>DASD CP owned (system areas)</td>
</tr>
<tr>
<td>Record name</td>
<td>Member name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>VMPERFT_6F</td>
<td>DRLRPT6F</td>
<td>SCSI device records</td>
</tr>
<tr>
<td>VMPERFT_6D</td>
<td>DRLRPT6D</td>
<td>Queued Direct Input Output (QDIO) support</td>
</tr>
<tr>
<td>VMPERFT_71</td>
<td>DRLRPT71</td>
<td>DASD SEEKs data</td>
</tr>
<tr>
<td>VMPERFT_A2</td>
<td>DRLRPTA2</td>
<td>SFS and BFS server data</td>
</tr>
<tr>
<td>VMPERFT_A4</td>
<td>DRLRPTA4</td>
<td>Multitasking users data</td>
</tr>
<tr>
<td>VMPERFT_A6</td>
<td>DRLRPTA6</td>
<td>TCP/IP server data</td>
</tr>
<tr>
<td>VMPERFT_A7</td>
<td>DRLRPTA7</td>
<td>TCP/IP links data</td>
</tr>
<tr>
<td>VMPERFT_A8</td>
<td>DRLRPTA8</td>
<td>Reusable server kernel summary data</td>
</tr>
<tr>
<td>VMPERFT_A9</td>
<td>DRLRPTA9</td>
<td>Linux application data</td>
</tr>
</tbody>
</table>
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.

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.

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.
Tivoli Decision Support for z/OS dialog options

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.

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.
Tivoli Decision Support for z/OS dialog options

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.

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.
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

<table>
<thead>
<tr>
<th>Tablespace Name</th>
<th>SPACE Allocated</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLSADSM</td>
<td>1584</td>
</tr>
<tr>
<td>DRLSC108</td>
<td>10080</td>
</tr>
<tr>
<td>DRLSCOM</td>
<td>20160</td>
</tr>
<tr>
<td>DRLSC01</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC02</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC03</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC04</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC05</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC06</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC07</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC08</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC09</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC10</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC11</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC12</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC13</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC14</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC15</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC16</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC17</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC18</td>
<td>1056</td>
</tr>
<tr>
<td>DRLSC19</td>
<td>1056</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

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 STOPSPACE 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
- DRLPURGECOD

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.
The report contains the following information:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Purge Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP_GEN_IP_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_TCP_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_API_CALLS_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_API_CALLS_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_UDP_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_ICMP_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_IP_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_TCP_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_UDP_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_ICMP_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_IP_W</td>
<td>DATE &lt; CURRENT DATE - 365 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_TCP_W</td>
<td>DATE &lt; CURRENT DATE - 365 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_UDP_W</td>
<td>DATE &lt; CURRENT DATE - 365 DAYS</td>
</tr>
<tr>
<td>TCP_GEN_ICMP_W</td>
<td>DATE &lt; CURRENT DATE - 365 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_CLIENT_T</td>
<td>DATE &lt; CURRENT DATE - 1 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_CLIENT_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_CLIENT_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_CLIENT_W</td>
<td>DATE &lt; CURRENT DATE - 365 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_SERVER_T</td>
<td>DATE &lt; CURRENT DATE - 1 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_SERVER_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_SERVER_D</td>
<td>DATE &lt; CURRENT DATE - 30 DAYS</td>
</tr>
<tr>
<td>TCP_FTP_SERVER_W</td>
<td>DATE &lt; CURRENT DATE - 365 DAYS</td>
</tr>
<tr>
<td>TCP_TN3270_CLNT_T</td>
<td>DATE &lt; CURRENT DATE - 1 DAYS</td>
</tr>
<tr>
<td>TCP_TN3270_CLNT_H</td>
<td>DATE &lt; CURRENT DATE - 7 DAYS</td>
</tr>
</tbody>
</table>

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

**Variables**
- Tablename.

List columns for a requested table with comments

<table>
<thead>
<tr>
<th>KEYS</th>
<th>KEYSEQ</th>
<th>NAME</th>
<th>COLTYPE</th>
<th>LENGTH</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>7</td>
<td>CORRELATION_ID</td>
<td>CHAR</td>
<td>12</td>
<td>Correl. ID value. From QWHCCV.</td>
</tr>
<tr>
<td>K</td>
<td>10</td>
<td>DB2_PLAN</td>
<td>CHAR</td>
<td>8</td>
<td>Plan name. From QWHCPLAN.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>BP32_DYN_PREFETCH</td>
<td>FLOAT</td>
<td>4</td>
<td>Num. of DYNAMIC PREFETCH requests</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>BP32_EXPANSIONS</td>
<td>FLOAT</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**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.
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
- 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:

<table>
<thead>
<tr>
<th>NAME</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS_DICTIONARY</td>
<td>CICS dictionary records. Used and maintained by the recordproc that</td>
</tr>
<tr>
<td>CICS_FIELD</td>
<td>CICS dictionary fields. Used to update the dictionary blocks in</td>
</tr>
<tr>
<td>DAY_OF_WEEK</td>
<td>This control table defines the day type to be returned by the</td>
</tr>
<tr>
<td></td>
<td>DAYTYPE</td>
</tr>
<tr>
<td>DB2_APPL_DIST_H</td>
<td>This table provides hourly statistics on DDF distributed address space</td>
</tr>
</tbody>
</table>

Figure 114. Example of List all tables with comment
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.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Object Type</th>
<th>Object Name</th>
<th>Member Name</th>
<th>Part Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSM</td>
<td>LOG</td>
<td>SMF</td>
<td>DRLLSMF</td>
<td>-</td>
<td>FLAG</td>
</tr>
<tr>
<td>CICSMON</td>
<td>LOG</td>
<td>SMF</td>
<td>DRLLSMF</td>
<td>-</td>
<td>FLAG</td>
</tr>
<tr>
<td></td>
<td>RECORD</td>
<td>SMF_110_1</td>
<td>DRLLSMF</td>
<td>-</td>
<td>FLAG</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>CICSA05</td>
<td>DRLOCI07</td>
<td>7 CMF GLOB &amp; ACCT PN86655</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CICSA07</td>
<td>DRLOCI07</td>
<td>7 CMF GLOB &amp; ACCT PN86655</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UPDATE</td>
<td>CICS_TRAN_USR_H</td>
<td>DRLTCTR</td>
<td>1 CMF BASIC ALTERED</td>
<td></td>
</tr>
<tr>
<td>CICSMOP</td>
<td>LOG</td>
<td>SMF</td>
<td>DRLLSMF</td>
<td>-</td>
<td>FLAG</td>
</tr>
<tr>
<td></td>
<td>RECORD</td>
<td>SMF_110_1</td>
<td>DRLLSMF</td>
<td>-</td>
<td>FLAG</td>
</tr>
<tr>
<td>CICSSTAP</td>
<td>LOG</td>
<td>SMF</td>
<td>DRLLSMF</td>
<td>-</td>
<td>FLAG</td>
</tr>
<tr>
<td></td>
<td>RECORD</td>
<td>SMF_110_2</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_02</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_07</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_08</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_10</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_10B</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_11</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_12</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_16</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_17</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_18</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_21</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_23</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_24</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_25</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_28</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_30</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_34</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_37</td>
<td>DRLR1102</td>
<td>-</td>
<td>P028635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_39</td>
<td>DRLR1102</td>
<td>-</td>
<td>P028635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_40</td>
<td>DRLR1102</td>
<td>-</td>
<td>P028635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_45</td>
<td>DRLR1102</td>
<td>-</td>
<td>P028635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_48</td>
<td>DRLR1102</td>
<td>-</td>
<td>P028635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_48_2</td>
<td>DRLR1102</td>
<td>-</td>
<td>P028635</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_54</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_61</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_63</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_66</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_76</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_81</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_85</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_87</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_2_94</td>
<td>DRLR1102</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMF_110_3</td>
<td>DRLR1103</td>
<td>-</td>
<td>P03356</td>
</tr>
<tr>
<td>RECPROC</td>
<td>DRL2CIST</td>
<td>DRLR1103</td>
<td>-</td>
<td>ALTERED</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>CICS_S_TCPIP_DP</td>
<td>DRLTS3P7</td>
<td>-</td>
<td>P03356</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CICS_S_TCPIP_TP</td>
<td>DRLTS3P7</td>
<td>-</td>
<td>P03356</td>
<td></td>
</tr>
</tbody>
</table>
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.
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 DRL1SQLXsql_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.
**REXX example of calling DRL1SQLX**

```rexx
/**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 */
sqlstmt = '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

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End

End
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>
<thead>
<tr>
<th>Analytics</th>
<th>Non-Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - DB2</td>
<td>DB2</td>
</tr>
<tr>
<td>Analytics - KPM CICS</td>
<td>Key Performance Metrics - CICS</td>
</tr>
<tr>
<td>Analytics - KPM DB2</td>
<td>Key Performance Metrics - DB2</td>
</tr>
<tr>
<td>Analytics - KPM z/OS</td>
<td>Key Performance Metrics – z/OS</td>
</tr>
<tr>
<td>Analytics - z/OS Performance</td>
<td>z/OS Performance Management (MVSPM)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Member name</th>
<th>Analytics Lookup table</th>
<th>non-Analytics Lookup table</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLTA2AP</td>
<td>A_DB2_APPLICATION</td>
<td>DB2_APPLICATION</td>
</tr>
<tr>
<td>DRLTA2AC</td>
<td>A_DB2_ACCUMAC</td>
<td>DB2_ACCUMAC</td>
</tr>
<tr>
<td>DRLTALUG</td>
<td>A_USER_GROUP</td>
<td>USER_GROUP</td>
</tr>
<tr>
<td>DRLTALKP</td>
<td>A_KPM_THRESHOLDS</td>
<td>KPM_THRESHOLDS</td>
</tr>
<tr>
<td>DRLTALW2</td>
<td>A_WORKLOAD2</td>
<td>MVS_WORKLOAD2_TYPE</td>
</tr>
<tr>
<td>DRLTALDA</td>
<td>A_DEVICE_ADDR</td>
<td>MVSPM_DEVICE_ADDR</td>
</tr>
<tr>
<td>DRLTALUT</td>
<td>A_UNIT_TYPE</td>
<td>MVSPM_UNIT_TYPE</td>
</tr>
<tr>
<td>DRLTALMI</td>
<td>A_MIPS</td>
<td>MVS_MIPS</td>
</tr>
<tr>
<td>DRLTALSP</td>
<td>A_SYSPLEX</td>
<td>MVS_SYSPLEX</td>
</tr>
<tr>
<td>DRLTALWL</td>
<td>A_WORKLOAD</td>
<td>MVS_WORKLOAD_TYPE</td>
</tr>
<tr>
<td>DRLTALTR</td>
<td>A_TIME_RES</td>
<td>MVSPM_TIME_RES</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Report name</th>
<th>Analytics Report ID</th>
<th>non-Analytics Report ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Buffer Pool Exceptions</td>
<td>ADB219</td>
<td>DB219</td>
</tr>
<tr>
<td>DB2 Buffer Pool Statistics, Detail</td>
<td>ADB215</td>
<td>DB215</td>
</tr>
<tr>
<td>DB2 Buffer Pool Statistics, Overview</td>
<td>ADB216</td>
<td>DB216</td>
</tr>
<tr>
<td>DB2 DBRMms Class 7,8 Times, Overview</td>
<td>ADB222</td>
<td>DB222</td>
</tr>
<tr>
<td>DB2 General Measure by Profile, Overview</td>
<td>ADB212</td>
<td>DB212</td>
</tr>
<tr>
<td>DB2 IDAA Statistics by Transaction, Detail</td>
<td>ADB244</td>
<td>DB244</td>
</tr>
<tr>
<td>DB2 Packages Class 7,8 Times, Overview</td>
<td>ADB221</td>
<td>DB221</td>
</tr>
<tr>
<td>DB2 Transaction Statistics, Detail</td>
<td>ADB204</td>
<td>DB204</td>
</tr>
<tr>
<td>DB2 Transaction Statistics, Overview</td>
<td>ADB205</td>
<td>DB205</td>
</tr>
</tbody>
</table>

The following table lists all the tables per Analytics component, and their equivalent non-Analytics component tables.
<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Analytics component table</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - z/OS</td>
<td>Table</td>
<td>A_PM_CF_I</td>
<td>MVSPM_CF_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_CF_LINK_I</td>
<td>MVSPM_CF_LINK_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_CF_PROC_I</td>
<td>MVSPM_CF_PROC_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_CF_REQ_I</td>
<td>MVSPM_CF_REQUEST_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_CF_CF_I</td>
<td>MVSPM_CF_TO_CF_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_XCF_MEMBER_I</td>
<td>MVSPM_XCF_MEMBER_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_XCF_PATH_I</td>
<td>MVSPM_XCF_PATH_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_XCF_SYS_I</td>
<td>MVSPM_XCF_SYS_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_OMVS_BUF_I</td>
<td>MVSPM_OMVS_BUF_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_OMVS_FILE_I</td>
<td>MVSPM_OMVS_FILE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_OMVS_GHFS_I</td>
<td>MVSPM_OMVS_GHFS_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_OMVS_HFS_I</td>
<td>MVSPM_OMVS_HFS_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_OMVS_KERN_I</td>
<td>MVSPM_OMVS_KERN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_OMVS_MOUNT_I</td>
<td>MVSPM_OMVS_MOUNT_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_CLUST_I</td>
<td>MVSPM_CLUSTER_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_CPU_I</td>
<td>MVSPM_CPU_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_CPUMT_I</td>
<td>MVSPM_CPUMT_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_ENQ_I</td>
<td>MVSPM_ENQUEUE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_LPAR_I</td>
<td>MVSPM_LPAR_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_SYS_I</td>
<td>MVSPM_SYSTEM_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_PROD_I</td>
<td>MVSPM_PROD_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_PRDINT_I</td>
<td>MVSPM_PROD_INT_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_SYS_MSU_I</td>
<td>MVSPM_LPAR_MSU_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_WL_GOAL_I</td>
<td>MVSPM_GOAL_ACT_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_WL_SERVED_I</td>
<td>MVSPM_WLM_SERVED_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_WL_STATE_I</td>
<td>MVSPM_WLM_STATE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_WL_WKLD_I</td>
<td>MVSPM_WORKLOAD_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_WL_WKLD2_I</td>
<td>MVSPM_WORKLOAD2_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_IO_DATASET_I</td>
<td>MVSPM_DATASET_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_IO_VOLUME_I</td>
<td>MVSPM_VOLUME_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_IO_LCU_I</td>
<td>MVSPM_LCU_IO_H</td>
</tr>
</tbody>
</table>

Table 11. Relationship of Analytics component table to non-Analytics component table
### Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Analytics component table</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - z/OS Performance</td>
<td>Table</td>
<td>A_PM_GS_BMF_I</td>
<td>MVSPM_BMF_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_CACHE_I</td>
<td>MVSPM_CACHE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_PAGEDS_I</td>
<td>MVSPM_PAGE_DS_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_PAGING_I</td>
<td>MVSPM_PAGING_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_STORAGE_I</td>
<td>MVSPM_STORAGE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_STORCLS_I</td>
<td>MVSPM_STORCLASS_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_SWAP_I</td>
<td>MVSPM_SWAP_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_GS_CACHESS_I</td>
<td>MVSPM_CACHE_ESS_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_VS_VLF_I</td>
<td>MVSPM_VLF_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_VS_CSASQA_I</td>
<td>MVSPM_VS_CSASQA_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_VS_PRIVATE_I</td>
<td>MVSPM_VS_PRIVATE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_VS_SUBPOOL_I</td>
<td>MVSPM_VS_SUBPOOL_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_CHAN_I</td>
<td>MVSPM_CHANNEL_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_HSCHAN_I</td>
<td>MVSPM_HS_CHAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_AP_I</td>
<td>MVSPM_DEVICE_AP_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_DEVICE_I</td>
<td>MVSPM_DEVICE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_FICON_I</td>
<td>MVSPM_FICON_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_RAID_I</td>
<td>MVSPM_RAID_RANK_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_ESSLNK_I</td>
<td>MVSPM_ESSLINK_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_ESSEXT_I</td>
<td>MVSPM_ESS_EXTENT_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_ESSRNK_I</td>
<td>MVSPM_ESS_RANK_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_DEV_PCIE_I</td>
<td>MVSPM_PCIE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_CRYP_PCI_I</td>
<td>MVSPM_CRYPTO_PCI_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_CRYP_CCF_I</td>
<td>MVSPM_CRYPTO_CCF_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_PM_APP_APPL_I</td>
<td>MVSPM_APPL_H</td>
</tr>
</tbody>
</table>
Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Analytics component table</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - DB2</td>
<td>Table</td>
<td>A_DB2_SYS_PARM_I</td>
<td>DB2_SYS_PARAMETER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DB_I</td>
<td>DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DB_BIND_I</td>
<td>DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DB_QIST_I</td>
<td>DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DB_SYS_I</td>
<td>DB2_SYSTEM_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_BP_I</td>
<td>DB2_BUFFER_POOL_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_USERTRAN_I</td>
<td>DB2_USER_TRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_UT_BP_I</td>
<td>DB2_USER_TRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_UT_SACC_I</td>
<td>DB2_USER_TRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_UT_IDAA_I</td>
<td>DB2_USER_TRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_IDAA_STAT_I</td>
<td>DB2_IDAA_STAT_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_IDAA_ACC_I</td>
<td>DB2_IDAA_ACC_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_IDAA_ST_A_I</td>
<td>DB2_IDAA_STAT_A_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_IDAA_ST_S_I</td>
<td>DB2_IDAA_STAT_S_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_PACK_I</td>
<td>DB2_PACKAGE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_SHR_BP_I</td>
<td>DB2_BP_SHARING_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_SHR_BPAT_I</td>
<td>DB2_BPAATTR_SHR_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_SHR_LOCK_I</td>
<td>DB2_LOCK_SHARING_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_SHR_INIT_I</td>
<td>DB2_SHARING_INIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_SHR_TRAN_I</td>
<td>DB2_USER_TRAN_SHAR_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DDF_I</td>
<td>DB2_USER_DIST_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_SYSTEM_I</td>
<td>DB2_SYSTEM_DIST_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_STORAGE_I</td>
<td>DB2_STORAGE_T</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>A_DB2_TRAN_IV</td>
<td>DB2_TRANSACTION_D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DATABASE_IV</td>
<td>DB2_DATABASE_T</td>
</tr>
</tbody>
</table>
Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Component table</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics – KPM DB2</td>
<td>Table</td>
<td>A_KD_UT_I</td>
<td>KPM_DB2_USERTRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_UT_BP_I</td>
<td>KPM_DB2_USERTRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_EU_I</td>
<td>KPM_DB2_ENDUSER_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_EU_BP_I</td>
<td>KPM_DB2_ENDUSER_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_PACKAGE_I</td>
<td>KPM_DB2_PACKAGE_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_IO_I</td>
<td>KPM_DB2_SYSTEM_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_TCBSRB_I</td>
<td>KPM_DB2_SYSTEM_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_LATCH_I</td>
<td>KPM_DB2_LATCH_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_BP_I</td>
<td>KPM_DB2_BP_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_BP_SHR_I</td>
<td>KPM_DB2_BP_SHR_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_ST,DBM_I</td>
<td>KPM_DB2_STORAGE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_ST,DST_I</td>
<td>KPM_DB2_STORAGE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_SYS_ST,COM_I</td>
<td>KPM_DB2_STORAGE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB_SYS_DB,WF_I</td>
<td>KPM_DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB_SYS_DB,EDM_I</td>
<td>KPM_DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB_SYS_DB,SET_I</td>
<td>KPM_DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB_SYS_DB,LOCK_I</td>
<td>KPM_DB2_LOCK_T</td>
</tr>
<tr>
<td>Analytics – KPM CICS</td>
<td>Table</td>
<td>A_KC_MON_TRAN_I</td>
<td>KPMC_MON_TRAN_H</td>
</tr>
</tbody>
</table>

IBM DB2 Analytics Accelerator
### Table 11. Relationship of Analytics component table to non-Analytics component table (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Analytics component table</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - KPM z/OS Table</td>
<td>A_KPM_EXCEPTION_I</td>
<td>KPM_EXCEPTION_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_JOB_INT_I</td>
<td>KPMZ_JOB_INT_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_JOB_STEP_I</td>
<td>KPMZ_JOB_STEP_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_LPAR_I</td>
<td>KPMZ_LPAR_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_STORAGE_I</td>
<td>KPMZ_STORAGE_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_WORKLOAD_I</td>
<td>KPMZ_WORKLOAD_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CHANNEL_I</td>
<td>KPMZ_CHANNEL_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CF_I</td>
<td>KPMZ_CF_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CF_STRUCT_I</td>
<td>KPMZ_CF_STRUCT_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CPUMF_I</td>
<td>KPMZ_CPUMF_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CPUMF1_I</td>
<td>KPMZ_CPUMF1_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CPUMF_PT_I</td>
<td>KPMZ_CPUMF_PT_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_CPUMF1_PT_I</td>
<td>KPMZ_CPUMF1_PT_T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A_KZ_SRM_WKLD_I</td>
<td>KPMZ_SRM_WKLD_T</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Component</th>
<th>View</th>
<th>Analytics component tables used in view</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - DB2</td>
<td>A_DB2_USERTRAN_IV</td>
<td>A_DB2_USERTRAN_I</td>
<td>DB2_USER_TRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_UT_BP_I</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_UT_SACC_I</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_UT_IDAA_I</td>
<td></td>
</tr>
<tr>
<td>Analytics - DB2</td>
<td>A_DB2_DATABASE_IV</td>
<td>A_DB2_DB_I</td>
<td>DB2_DATABASE_T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DB_BIND_I</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_DB2_DB_QIST_I</td>
<td></td>
</tr>
<tr>
<td>Analytics - KPM DB2</td>
<td>A_KD_USERTRAN_IV</td>
<td>A_KD_UT_I</td>
<td>KPM_DB2_USERTRAN_H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A_KD_UT_BP_I</td>
<td></td>
</tr>
</tbody>
</table>

Chapter 18. Using the IBM DB2 Analytics Accelerator 301
Table 12. Relationship of Analytics component tables used in view to non-Analytics component tables used in view (continued)

<table>
<thead>
<tr>
<th>Component</th>
<th>View</th>
<th>Analytics component tables used in view</th>
<th>Equivalent to non-Analytics component table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics - KPM DB2</td>
<td>A_KD_ENDUSER_IV</td>
<td>A_KD_EU_I, A_KD_EU_BP_I</td>
<td>KPM_DB2_ENDUSER_H</td>
</tr>
<tr>
<td>Analytics - KPM DB2</td>
<td>A_KD_SYSTEM_IV</td>
<td>A_KD_SYS_IO_I, A_KD_SYS_TCBSRB_I</td>
<td>KPM_DB2_SYSTEM_T</td>
</tr>
<tr>
<td>Analytics - KPM DB2</td>
<td>A_KD_STORAGE_IV</td>
<td>A_KD_SYS_ST_DBM_I, A_KD_SYS_ST_DST_I, A_KD_SYS_ST_COM_I</td>
<td>KPM_DB2_STORAGE_T</td>
</tr>
<tr>
<td>Analytics - KPM DB2</td>
<td>A_KD_DATABASE_IV</td>
<td>A_DB_SYS_DB_WF_I, A_DB_SYS_DB_EDM_I, A_DB_SYS_DB_SET_I</td>
<td>KPM_DB2_DATABASE_T</td>
</tr>
</tbody>
</table>

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(DSNTIxx) -
  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 = "xxxxxxxx"
  "xxxxxxxx": 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
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**

<table>
<thead>
<tr>
<th>SDRLCNTL member name</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLJA2DA</td>
<td>Analytics - DB2</td>
</tr>
<tr>
<td>DRLJAPMA</td>
<td>Analytics - z/OS Performance</td>
</tr>
<tr>
<td>DRLJAKCA</td>
<td>Analytics - KPM CICS</td>
</tr>
<tr>
<td>DRLJAKDA</td>
<td>Analytics - KPM DB2</td>
</tr>
<tr>
<td>DRLJAKZA</td>
<td>Analytics - KPM z/OS</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>SDRLCNTL member name</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLJAPMK</td>
<td>Analytics - z/OS Performance</td>
</tr>
<tr>
<td>DRLJAKDK</td>
<td>Analytics - KPM DB2</td>
</tr>
<tr>
<td>DRLJAKZK</td>
<td>Analytics - KPM z/OS</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>SDRLCNTL member name</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLJA2DD</td>
<td>Analytics - DB2</td>
</tr>
<tr>
<td>DRLJAPMD</td>
<td>Analytics - z/OS Performance</td>
</tr>
<tr>
<td>DRLJAKCD</td>
<td>Analytics - KPM CICS</td>
</tr>
<tr>
<td>DRLJAKDD</td>
<td>Analytics - KPM DB2</td>
</tr>
<tr>
<td>DRLJAKZD</td>
<td>Analytics - KPM z/OS</td>
</tr>
</tbody>
</table>

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_useraat 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 Tivoli Decision Support for z/OS: Administration Guide and Reference

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

<table>
<thead>
<tr>
<th>SDRLCNTL member name</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLJA2DE</td>
<td>Analytics - DB2</td>
</tr>
<tr>
<td>DRLJAPME</td>
<td>Analytics - z/OS Performance</td>
</tr>
<tr>
<td>DRLJAKCE</td>
<td>Analytics - KPM CICS</td>
</tr>
<tr>
<td>DRLJAKDE</td>
<td>Analytics - KPM DB2</td>
</tr>
<tr>
<td>DRLJAKZE</td>
<td>Analytics - KPM z/OS</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>SDRLCNTL member name</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLJA2DR</td>
<td>Analytics - DB2</td>
</tr>
<tr>
<td>DRLJAPMR</td>
<td>Analytics - z/OS Performance</td>
</tr>
<tr>
<td>DRLJAKCR</td>
<td>Analytics - KPM CICS</td>
</tr>
<tr>
<td>DRLJAKDR</td>
<td>Analytics - KPM DB2</td>
</tr>
<tr>
<td>DRLJAKZR</td>
<td>Analytics - KPM z/OS</td>
</tr>
</tbody>
</table>

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/](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:

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/customercare/sas/f/handbook/home.html](http://www-304.ibm.com/support/customercare/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:

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](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](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](https://techsupport.services.ibm.com/ssr/login).


- 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](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](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](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](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.
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.

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.

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.

update definitions
Instructions for entering data into DB2 tables from records of different types or from other DB2 tables.

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
- Resource Accounting for z/OS, SH19-4495
- AS/400 System Performance Feature Guide and Reference, SH19-4019
- CICS Performance Feature Guide and Reference, SH19-6820
- Distributed Systems Performance Feature Guide and Reference, SH19-4018
- Guide to Reporting, SH19-6842
- IMS Performance Feature Guide and Reference, SH19-6825
- Language Guide and Reference, SH19-6817
- Messages and Problem Determination, SH19-6902
- Network Performance Feature Installation and Administration, SH19-6901
- Network Performance Feature Reference, SH19-6822
- Network Performance Feature Reports, SH19-6821
- Resource Accounting, SH19-6818
- System Performance Feature Guide, SH19-6819
- System Performance Feature Reference Volume I, SH19-4494
- System Performance Feature Reference Volume II, 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
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
    DRLREIN1 27
    generic logon procedure 27
    JCL sample jobs 35
  customizing Usage and Accounting Collector
    execute DRLNINIT 42

D

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
    monitoring 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.SDRLDEFS 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
exception (continued) 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
DRLLENI1 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
when QMF is used 56
disability 309
DISPlay RECORD commands 279
DISPlay REPORT commands 279
DISPlay TABLE commands 279
DISPlay table_name commands 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 63
DRL_LOCAL.EXEC 27
DRL_LOCAL.REPORTS 113
DRL_LOCAL.USER.DEFS 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
DRLBIN 279
DRLBIN job 22
DRLJDBIIP database initialization job 26
DRLJDBIP job 279
DRLJDMC setting the parameters for job 218
DRLJDMC collect job parameters it uses 215
DRLJDMC job step, using 206
sample job 207
setting the parameters for 208
DRLJDMC sample job 207
DRLJFURG purge job 104
DRLJFRT report format table 89
DRLJRUNS RUNSTATS job 108
DRLKEYS view on DB2 catalog 241
DRLLLDM_COLLECTSTMT system table 226
DRLLLDM_LOGDATASETS system table 227
DRLLOGDATASETS system table 227
DRLLOGDATASETS system table 92
DRLLOGS system table 228
DRLLOGIN 42
DRLNL_JOB2 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
DRLREPORTS system table 230
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
DRLPROCINPUT 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
DRLUPDATING system table 231
DRLUPDATING system table 231
DRLUPDATING system table 231
DRLUPDATING system table 231
DRLUPDATING system table 231
DRLUSER_GROUPREPS system table 229
DRLUSER_GROUPS system table 225
DRLUSER_REPOR system table 225
DRLUSER_REPOR system table 229
DRLUSER_REPOR system table 230
DRLUSER_SEARCHES system table 225
DRLUSER_SEARCHA system table 225
DRLUSER_SEARCHA system table 225
DRLUSER_SEARCHA system table 225
DRLREPOR 241
DRLREPOR 241
DRLREPOR 241
DRLREPOR 241
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPOR system table 230
DRLREPORT_ATTR system table 238
DRLREPORT_COLUMNS system table 239
DRLREPORT_QUERIES system table 239
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
exceptional_xi
excluding an object from a component installation 139
exporting table data to an IXF file 172
information centers, searching for problem resolution 311
installation base product and feature installation 17, 36
description of DRLEIN11 variables 64
DRLEIN11 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 ISPFPROFILE 27
ISPF command 280
ISPF editor editing a table 167
IXF exporting table data to 172
IXF file importing contents to a table 171

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.SDRLLOAD 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
LOxate 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
making table space parameter changes that do not require offline or batch action 195
making changes to an index space 194

M
making changes to an index space 110
maintaining database security 110
log definitions (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
DB2 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

N
naming convention for members of DRL1825DRLDEFS 81
naming convention for members of DRL1825DRLRENU 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
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
DB2 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 DRLNJ0B3 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
DRLLEDMC 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
system tables and views (continued)
  DRLTABLES 241
  DRLTABLESPACE 241
  DRLUPDATECOLS 231
  DRLUPDATEDISTR 231
  DRLUPDATEGTS 231
  DRLUPDATES 232
  DRLUSER_GROUPREPS 241
  DRLUSER_GROUPS 241
  DRLUSER_REPORTATTR 241
  DRLUSER_REPORTCOLS 242
  DRLUSER_REPORTQRY 242
  DRLUSER_REPORTS 242
  DRLUSER REPORTTEXT 242
  DRLUSER_REPORTVARS 242
  DRLUSER_SEARCHATTR 242
  DRLUSER_SEARCHES 242
  DRLVIEW 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
  IFO 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
  DRLMM_COLLECTSTMT 226
  DRLMM_LOGDATASETS 227
  DRLMMLOGDATASETS 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
  DRLREPORTPROCINPUT 230
  DRLSEARCH_ATTR 240
  DRLSEARCHES 240
  DRLSECTIONS 230
  DRLTABLEAUTH 241
  DRLTABLEPART 241
  DRLTABLES 241
  DRLTABLESPACE 241
  DRLUPDATECOLS 231
  DRLUPDATEDISTR 231
  DRLUPDATEGTS 231
  DRLUPDATES 232
  DRLUSER_GROUPREPS 241
  DRLUSER_GROUPS 241
  DRLUSER_REPORTATTR 241
  DRLUSER_REPORTCOLS 242
  DRLUSER_REPORTQRY 242
  DRLUSER_REPORTS 242
  DRLUSER REPORTTEXT 242
  DRLUSER_REPORTVARS 242

Index 329
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 36, 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 DRLJLDML job step 206
utility
Purge 104
Remove/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
diag system tables 235
DRLCHARTS 235
DRLCOLUMNS 241
DRLCOMP_OBJECTS 236
DRLCOMP_PARTS 236
DRLCOMPONENTS 236
DRLGROUPS 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
DRLUPDATEETS 231
DRLUPDATES 232
DRLUSER_GROUPREPS 241
DRLUSER_GROUPS 241
DRLUSER_REPORTATTR 241
DRLUSER_REPORTCOLS 242
DRLUSERREPORTQRY 242
DRLUSERREPORTS 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

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

z

z/VM Performance Toolkit
record definitions 270